A Reinforcement Learning and Behavior Cloning Based Decision-Making System for Autonomous Driving and Real-Vehicle Deployment

Xin Xing[1], Jannes Bikker[2] and Sebastian Ohl[3]

***Abstract*—Decision-making is a critical part of autonomous driving systems with robust and adaptive policies in complex traffic scenarios. Although Reinforcement Learning (RL) and Behavior Cloning (BC) have performed promising, both have limitations and apply separately (i.e., unsafe exploration in RL, BC distribution shift). In this contribution, we propose a two-stage decision-making framework utilizing Proximal Policy Optimization (PPO) and BC for maneuver-based Decision-making. Based on the two-step training, an LSTM-based actor–critic network is first trained using PPO and then fine-tuned by real-world driving. A behavior-level representation based on Adaptive Cruise Control (ACC) and Autonomous Emergency Braking (AEB) evaluations is proposed for the benefit of learning efficiency and interpretability. Experiments in simulation and real-vehicle scenarios show that the proposed framework can achieve fast convergence, stable training behavior, and reliable performance in safety-critical scenarios. Real-world test results demonstrate that the system can recognize emergencies and brake on a timely basis, demonstrating the utility of the proposed system in practical autonomous driving applications.**

***Index Terms*—Autonomous Driving, Decision-Making, Reinforcement Learning, Behavior Cloning, PPO, LSTM, Simulation-to-real Transfer.**

I. INTRODUCTION

Decision-making is a core component of autonomous driving systems, responsible for selecting appropriate maneuvers in complex and dynamic traffic environments [1]. Autonomous vehicles must continuously balance safety, efficiency, and comfort while interacting with surrounding traffic participants. Traditional rule-based and modular control approaches have shown limited adaptability in highly uncertain scenarios, motivating the adoption of data-driven learning methods [2].

Behavior Cloning (BC), as a form of imitation learning, enables vehicles to learn driving policies directly from human demonstrations [3], [4]. By leveraging large-scale driving datasets, BC can produce human-like behavior and achieve strong performance in nominal scenarios. However, BC suffers from distribution shift and compounding errors, as the learned policy is only exposed to expert data during training. When encountering unseen states, BC models often fail to recover, limiting their robustness in real-world environments [4], [5].

[1]Xin Xing is with Faculty of Electrical Engineering and Information Technology, Ostfalia University of Applied Sciences, 38302 Wolfenbuettel, Germany xi.xing@ostfalia.de

[2]Jannes Bikker is with Faculty of Computer Science/IT, Ostfalia University of Applied Sciences, 38302 Wolfenbuettel, Germany ja.bikker@ostfalia.de

[3]Sebastian Ohl is with Faculty of Electrical Engineering and Information Technology, Ostfalia University of Applied Sciences, 38302 Wolfenbuettel, Germany s.ohl@ostfalia.de

Reinforcement Learning (RL) provides an alternative paradigm by optimizing long-term objectives through interaction with the environment. RL has been successfully applied to various driving tasks, including lane following, adaptive cruise control, and lane changing [6], [7]. In our previous studies, RL-based maneuver decision frameworks demonstrated effective performance in simulation [8], [9]. However, RL methods face challenges such as low sample efficiency, unsafe exploration, and reward design complexity, which restrict their direct application in safety-critical systems [4].

Recent studies have shown that integrating BC and RL can address the limitations of each individual approach. BC provides a strong prior and improves learning efficiency, while RL enables closed-loop optimization and adaptation to novel situations. Another approach is to first train the policy using RL and then fine-tune it using BC. Hybrid frameworks, such as demonstration-regularized RL and learning from demonstrations, have achieved improved safety and robustness in challenging driving scenarios. Notably, Waymo's BC-SAC framework showed that combining imitation and reinforcement learning significantly reduces safety-critical events [5].

Another major challenge lies in transferring learned policies from simulation to real vehicles. Differences in sensor noise, vehicle dynamics, and environmental conditions often lead to performance degradation. Techniques such as domain randomization, affordance-based representations, and hierarchical architectures have been proposed to mitigate the sim-to-real gap [7], [9]. Our previous work validated that maneuver-based RL policies trained in the Simulation Webots could be deployed on real vehicles with satisfactory performance [8], [9], [10].

In this work, we propose a unified decision-making framework that integrates BC and deep reinforcement learning for longitudinal maneuver decision-making in autonomous driving, with a particular focus on real-vehicle deployment. The proposed system applies reinforcement learning for policy initialization and expert demonstrations for fine-tuning. Extensive evaluations in simulation and real-world experiments demonstrate the effectiveness and robustness of the proposed approach.

II. RELATED WORK

A. Imitation Learning for Autonomous Driving

Imitation Learning (IL), particularly Behavior Cloning, has long been used to train autonomous driving policies by mimicking human drivers [3]. From early systems like ALVINN

to NVIDIA's DAVE-2, research has demonstrated that neural networks can map visual inputs to control commands. Modern BC approaches, combining deep convolutional networks with large-scale datasets, have achieved strong performance in lane-keeping and basic navigation tasks [2]. Techniques such as Conditional Imitation Learning have further enhanced adaptability in complex scenarios.

Recent studies show that vision-based BC models still maintain high accuracy and real-time performance in real-world environments [2], [5]. However, their effectiveness heavily depends on the training distribution and they are prone to accumulating errors in out-of-distribution settings. To address this, methods such as data augmentation, noise injection, and imitation regularization have been proposed to improve robustness [3]. However, pure BC still struggles with long-tail scenarios and complex decision-making problems, and is therefore increasingly combined with reinforcement learning [7].

B. Reinforcement Learning for Driving Decision-Making

RL treats autonomous driving as a sequential decision-making problem, learning optimal policies through a reward mechanism. Existing studies have applied algorithms such as DQN, PPO, SAC, and DDPG to tasks like lane changing, merging, and longitudinal control, demonstrating their advantages in continuous action spaces [6], [7].

In our previous work, we employed RL for Adaptive Cruise Control (ACC) and Autonomous Emergency Braking (AEB), using high-level maneuver modeling to enhance training stability and safety [8]. Experimental results show that DDPG tends to favor smooth and efficient driving, while PPO places more emphasis on safety [9].

Furthermore, incorporating mechanisms such as LSTM, GRU, or temporal attention can improve the policy's ability to model temporal dependencies, thereby enhancing robustness in dynamic environments [10], [11].

C. Hybrid Imitation and Reinforcement Learning

At present, hybrid approaches based on BC to RL represent an emerging paradigm in autonomous driving. Traditional hybrid methods either pretrain RL models via imitation learning, fine-tune them using reinforcement signals, or using imitation priors to regularize RL to guide the system toward safe and efficient policies [5], [7].

In one case, RL is first used to learn a powerful expert policy, which is distilled or fine-tuned via imitation learning into a deployable policy [5]. Several recent studies adopt this RL-to-IL (teacher–student) paradigm, in which an RL expert policy is trained first and then transferred through supervised imitation learning [4]. This approach leverages RL to discover effective driving behaviors, while imitation learning transfers those behaviors to models suitable for deployment under limited sensing or computational resources.

Zhang et al. proposed a two-stage framework in which an RL agent is first trained as a high-performance expert in the CARLA simulator, and then a supervised imitation learning agent is trained to mimic the RL expert's behavior using monocular camera input [4]. The RL coach provides dense supervision, enabling the student policy to achieve better generalization. In addition, Huang et al. introduced a method in which an imitation expert policy obtained via BC is used to regularize a deep RL agent [6]. The imitation component is further employed during fine-tuning to constrain and refine the RL policy, effectively embedding supervised objectives into the reinforcement learning framework.

III. Methodology

This study proposes a hybrid training framework that integrates PPO and BC to learn a robust longitudinal decision-making policy for discrete switching between ACC and AEB modes. As shown in Fig. 1, the overall pipeline consists of (i) simulation-based PPO pretraining and (ii) real-data-driven BC fine-tuning. During deployment, the trained LSTM-based policy receives behavior-level observations and outputs discrete control decisions in real time.

A. System Overview

The proposed system follows a two-stage learning strategy. First, an LSTM-based actor-critic policy is pretrained via reinforcement learning in a simulated environment to acquire fundamental longitudinal control behaviors and safety-aware decision patterns. Second, the pretrained policy is adapted to real-world driving conditions using supervised behavior cloning on real driving data. This design combines the exploration capability of reinforcement learning with the stability and realism of supervised learning, thereby reducing the simulation-to-reality gap.

B. PPO-Based Reinforcement Learning Pretraining

In the pretraining stage, the policy is optimized with PPO through interactions with the Webots environment [8], [12]. PPO is a policy-gradient method that stabilizes learning by restricting the policy update magnitude using a clipped surrogate objective. The overall PPO objective used in this work is defined as

$$L_{\mathrm{PPO}} = L_{\mathrm{clip}} + c_1 L_V, \tag{1}$$

where L_{clip} is the clipped surrogate objective and L_V is the value-function regression loss. The scalar c_1 balances the critic loss contribution.

The clipped objective is given by

$$L_{\mathrm{clip}} = \mathbb{E}[\min(r_t(\theta)A_t,\ \mathrm{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\, A_t)], \tag{2}$$

with the probability ratio

$$r_t(\theta) = \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\mathrm{old}}}(a_t \mid s_t)}, \tag{3}$$

where π_θ and $\pi_{\theta_{\mathrm{old}}}$ denote the updated and previous policies, respectively. A_t is the advantage estimate computed using Generalized Advantage Estimation (GAE), and ϵ is the clipping threshold. The clipping mechanism prevents overly large policy updates and improves training stability.

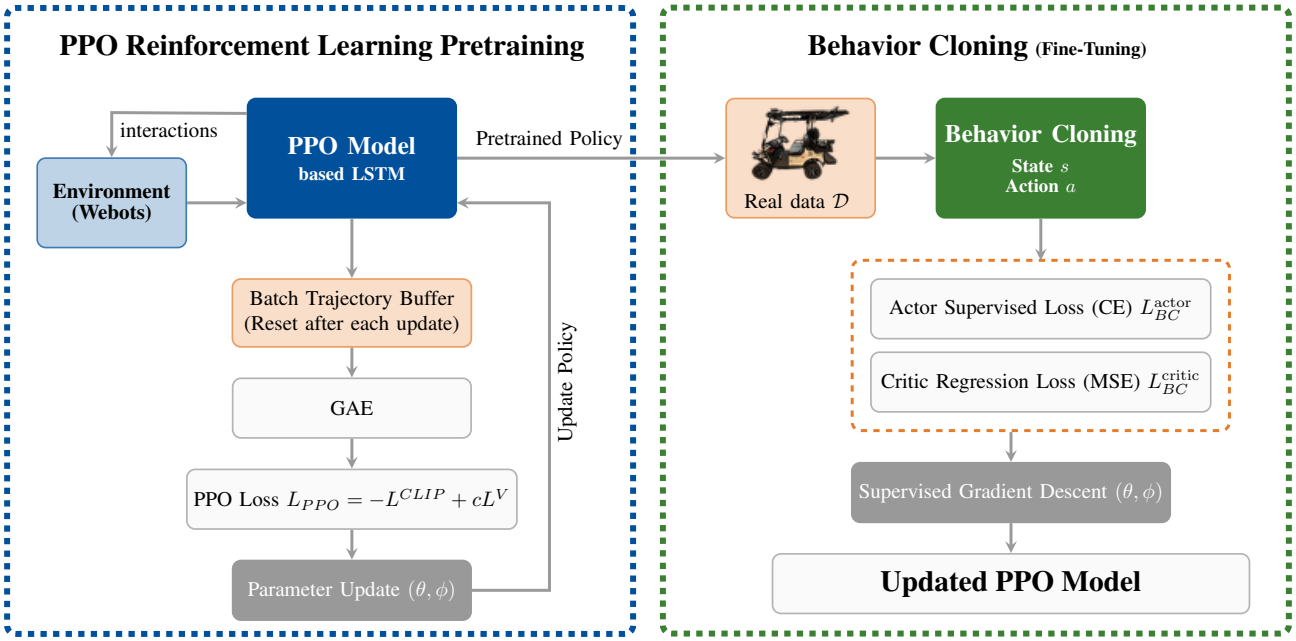


Fig. 1. Reinforcement Learning Pretraining and Behavior Cloning

The value-function loss is defined as

$$L_V = \mathbb{E}\Big[\big(V_\phi(s_t) - V_t^{\text{target}}\big)^2\Big], \tag{4}$$

where $V_\phi(s_t)$ is the critic value estimate with parameters ϕ, and V_t^{target} is the target return computed from the collected trajectories (e.g., via GAE).

The policy network employs an LSTM backbone to capture temporal dependencies in longitudinal driving. During training, trajectories are stored in a batch buffer and the actor–critic parameters (θ, ϕ) are updated after each optimization cycle [10].

C. Behavior Cloning Fine-Tuning

To further improve the generalization capability and robustness of the learned policy in real-world driving scenarios, a BC-based fine-tuning stage is introduced on top of the pretrained PPO model. This stage leverages real driving data to refine the policy in a supervised manner, enabling it to better align with human driving behaviors and mitigate the performance degradation caused by the simulation-to-reality gap. In addition to mimicking expert behaviour, the BC fine-tuning stage also serves to correct biased or unstable behaviours learned during simulation-based PPO pretraining. Specifically, the BC phase utilizes a dataset $\mathcal{D}$ consisting of expert demonstrations collected from human drivers or high-confidence autonomous systems, where each sample contains state–action pairs (s_t, a_t^{exp}). The Actor network is optimized by maximizing the log-likelihood of expert actions under the current policy distribution, resulting in the following cross-entropy loss:

$$L_{BC}^{\text{actor}} = \mathbb{E}_{(s_t, a_t^{\text{exp}})\sim\mathcal{D}}\left[-\log \pi_\theta(a_t^{\text{exp}} \mid s_t)\right]. \tag{5}$$

In addition, the critic is refined using a regression loss:

$$L_{BC}^{\text{critic}} = \mathbb{E}_{s_t\sim\mathcal{D}}\left[\big(V_\phi(s_t) - y_t\big)^2\right], \tag{6}$$

where $y_t = \sum_{k=0}^{T-t} \gamma^k r_{t+k}$ denotes the Monte-Carlo return computed from expert trajectories. Both networks are updated via supervised gradient descent to align the policy with real driving behaviors and mitigate the simulation-to-reality gap [13].

During training, both the Actor and Critic networks are jointly updated via supervised gradient descent. This BC-based fine-tuning stage serves as an effective regularization mechanism that corrects biased behaviors learned in simulation and adapts the policy to real-world dynamics and environmental uncertainties.

D. Training and Deployment Architecture

The complete training and deployment pipeline integrates simulation-based reinforcement learning, real-data-driven fine-tuning, and behavior-level perception.

Instead of directly using raw sensor measurements as the reinforcement learning state, the proposed system introduces an intermediate behavior evaluation layer, as shown in Fig. 2. During both training and deployment, raw sensor data are first processed by ACC and AEB behavior evaluators, which extract high-level metrics related to driving intention, comfort, and safety. Applicability shows if a behaviour is possible in the current driving situation. Desire shows if the current driving objective is preferred. Comfort shows how smooth manoeuvres are. Risk estimates potential collision or safety hazards.

Specifically, an 8-metric observation vector is constructed as

$$\mathbf{o}_t = [A_{acc}, D_{acc}, C_{acc}, R_{acc}, A_{aeb}, D_{aeb}, C_{aeb}, R_{aeb}]_t, \tag{7}$$

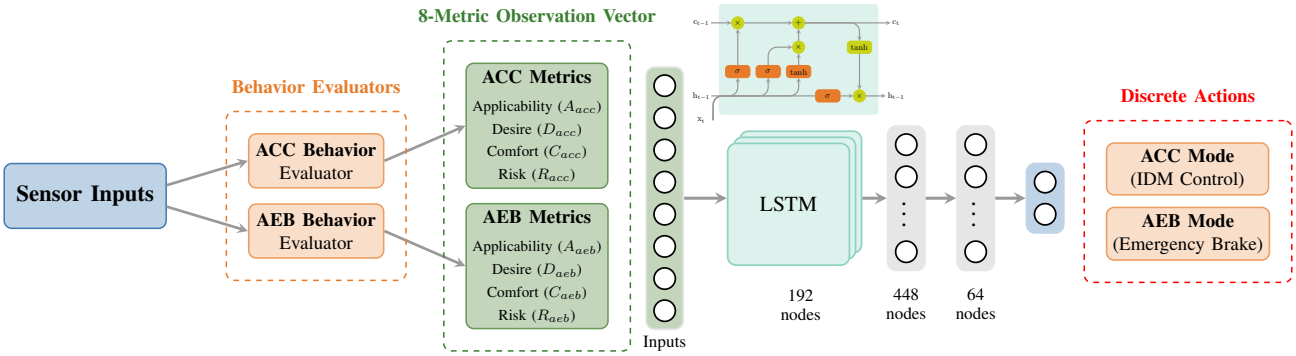


Fig. 2. Training and Deployment Architecture

where A, D, C, and R denote applicability, desire, comfort, and risk indicators for ACC and AEB modes, respectively.

This behavior-level representation serves as the state input to the reinforcement learning agent. Compared with raw sensor inputs, the proposed abstraction reduces input dimensionality, improves semantic interpretability, and enhances learning efficiency.

During training, the PPO–LSTM model interacts with the simulation environment using the observation vector $\mathbf{o}_t$ as the state input and outputs a discrete action $a_t \in \{\text{ACC}, \text{AEB}\}$. The environment executes the selected control mode and returns the next observation and reward, forming a closed-loop learning process. PPO optimization is performed using the objectives in Eqs. (1)–(4).

After PPO pretraining, the learned policy is transferred to the BC module and fine-tuned using real driving data with supervised losses in Eqs. (5)–(6), yielding the final updated model.

During deployment, onboard sensors continuously collect environmental and vehicle-state information. These measurements are transformed into behavior metrics by the evaluation modules and assembled into observation vectors. The trained LSTM policy receives a sequence of observation vectors and outputs real-time switching decisions between ACC (IDM-based control) and AEB (emergency braking) [8].

This hierarchical architecture decouples low-level perception from high-level decision-making, enabling robust learning, efficient training, and reliable real-world operation.

IV. Experiments and Results

A. System Overview

The experimental vehicle platform is designed to support systematic testing and evaluation of autonomous driving decision-making algorithms under real-world conditions. The system follows a centralized architecture integrating multi-modal perception, high-performance onboard computing, and reliable in-vehicle networking, as shown in Fig. 3.

The perception subsystem consists of a stereo camera, a MEMS solid-state LiDAR, and three 32-channels LiDARs, which together provide complementary visual and geometric information of the surrounding environment. To ensure precise

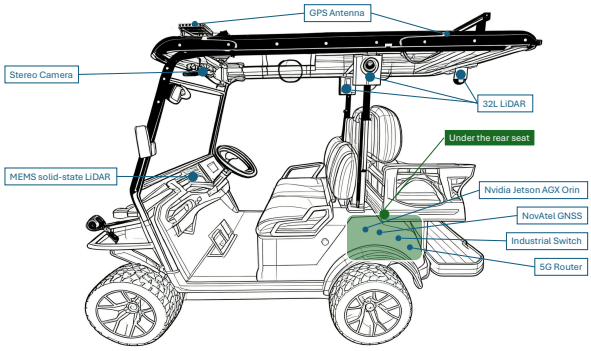


Fig. 3. Vehicle Hardware Overview

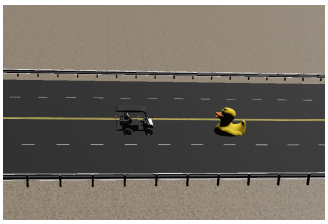
global localization and time synchronization, a NovAtel GNSS receiver is combined with a roof-mounted GPS antenna, providing position, velocity, and reference timing information. All sensor data are transmitted via an industrial Ethernet switch to centralized computing units, namely the NVIDIA Jetson AGX Orin. A 5G router is integrated into the onboard network, so that the GNSS system can provide the precise localization information by using the Real-Time Kinematic (RTK).

This embedded high-performance platform executes the main software pipeline, including sensor preprocessing, multi-sensor fusion, localization, decision-making and controlling. To support experimental management, all components are running in a K3S cluster [14].

B. Experimental Procedure and Data Collection

The experimental evaluation of the proposed reinforcement learning-based decision-making system was conducted using the vehicle platform described in the previous section. The overall experimental procedure consisted of three main stages: simulation-based pretraining, enhanced simulation validation, and real-world data collection.

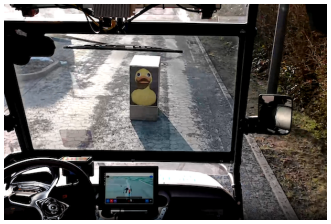
In the first stage, PPO-LSTM based models are trained in a dedicated simulation environment in Webots as shown in Fig. 4(a). This environment is configured to provide fundamental traffic scenarios, enabling efficient large-scale training and stable policy convergence. During this phase, multiple candidate models are obtained, and the model with the most



(a) Training Environment

(b) Simulation Environment

(c) Real-world External View



(d) Real-world Onboard View

Fig. 4. Training, simulation, and real-world evaluation scenes.

competitive performance is selected as the baseline model for further experiments.

In the second stage, additional datasets are collected using an enhanced simulation environment, as shown in Fig. 4(b). Compared with the original training environment, this simulator is configured to more closely approximate real-world driving conditions. Specifically, it incorporates higher traffic density, more diverse road topologies, and various dynamic obstacles. These modifications enabled the trained policy to be evaluated under more complex and challenging scenarios, thereby improving its generalization capability before real-world deployment.

In addition, real-world driving experiments are carried out on the proposed experimental vehicle platform as shown in Figs. 4(c) and 4(d). The synchronized multi-modal sensor data, vehicle state data, and environment features were recorded during field tests. These data included sensor data and RL model inputs.

The real-world and improved simulation data are used for behavior cloning training and performance evaluation. Figure 5 shows the training during behavior cloning. Both the Actor and Critic losses drop quickly in the early epochs, suggesting that the supervised learning is well converged. In particular, the Actor loss drops to near zero after a few iterations, while the Critic loss drops steadily in general, as the value estimation accuracy increases. The classification accuracy also increases from 68.99% in the first epoch to nearly 100% from the second epoch onward and stays stable throughout training.

C. Test with real vehicle

To test the effectiveness of our optimized model, real-vehicle experiments are conducted in real-world driving conditions. We conduct multiple tests in multiple conditions including obstacle-free navigation to the destination, static obstacles along the route and obstacle emergence in front

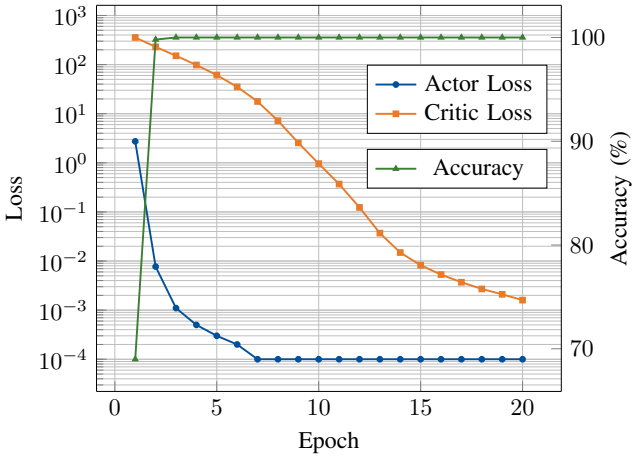


Fig. 5. Loss and Accuracy During Behavior Cloning

of the autonomous vehicle. This section presents the results corresponding to the third scenario in the following.

The development of autonomous vehicle velocity v_{AV}, distance d, and relative velocity v_{RV} between autonomous vehicle and obstacle is shown in Fig. 6. A stable after-condition is obtained with a relatively large distance and a small relative velocity, or without relative velocity if there is no detected target obstacle. If the sudden obstacle appears in front of the vehicle, the distance between the vehicles is rapidly reduced, with large variation of both vehicle speeds (i.e. collision risk). The system then slows down and eventually is able to stabilize the distance and velocity. This is a fundamental transition from normal driving - ACC to emergency braking - AEB. For safety considerations during field tests, the maximum vehicle velocity is limited to 10 km/h.

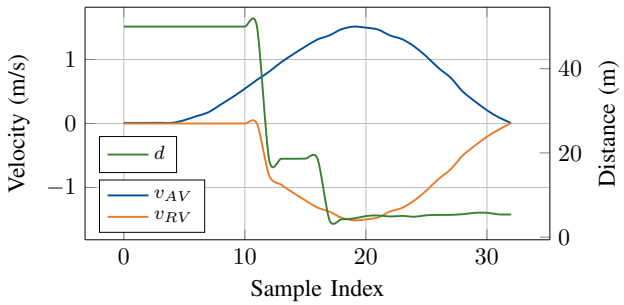


Fig. 6. Sensor data: autonomous vehicle's velocity v_{AV}, relative velocity v_{RV} and distance d.

Figure 7 provides further insights into the evaluation of ACC and AEB performance. ACC and AEB scores are computed for different aspects (applicability, desire, comfort and risk) and last active behavior. ACC scores are more important in the early stages and active behavior equals 1, indicating ACC activation. As distance between vehicle and obstacle decreases and risk increases, AEB scores become rapidly and highly important in terms of applicability and risk. This gives the opportunity to switch from ACC to AEB at high frequency. In the late stages, scores for both behaviors match, so that they

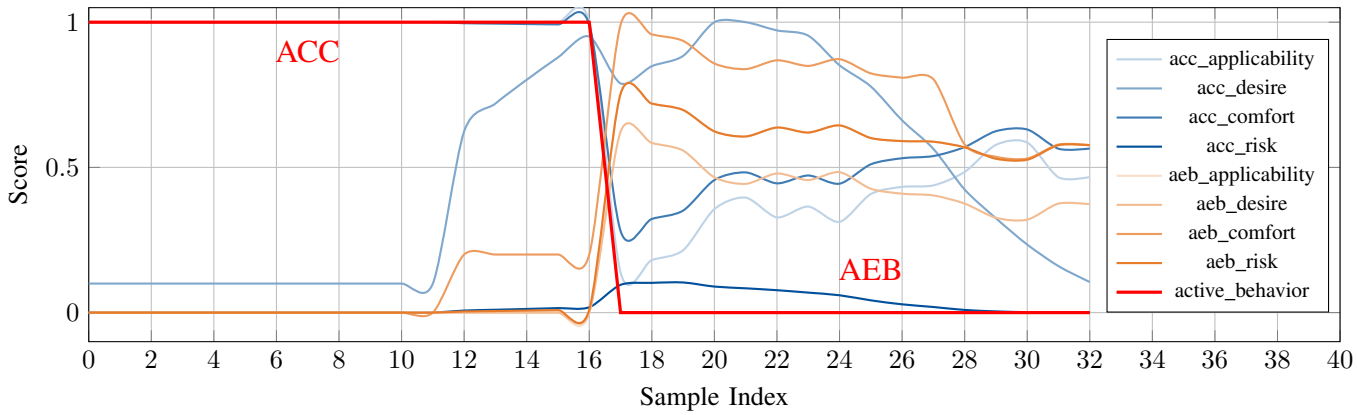


Fig. 7. ACC/AEB metrics as model inputs and active behavior as model output.

are back to equilibrium.

Overall, the results indicate that the model can adapt quickly and reliably to critical driving situations, i.e., safe and stable decisions can be made while autonomous decisions can have a sustained performance. BC-tuned PPO-LSTM has been able to detect AEB scenarios reliably and perform timely emergency braking in real driving experiments.

V. CONCLUSIONS

In this paper we presented a hybrid decision-making framework that combines reinforcement learning and behavior cloning for longitudinal control of self-driving systems. To combine PPO pretraining with supervised fine-tuning on real driving data, our approach utilizes the exploration advantage of reinforcement learning as well as the stability of imitation learning. Furthermore, with the introduction of behavior-level representations, our technique can effectively learn while maintaining semantic interpretability.

Experimental results from simulation and real-vehicle tests demonstrated that our approach achieves fast convergence and stable training and reliable performance in safety-critical situations. In particular, the proposed system is able to detect emergency braking scenarios and switch the behavioral switches between ACC and AEB modes efficiently. These results demonstrate the effectiveness of our proposed approach to reduce the simulation-to-reality gap and enhance the deployment reliability. Future work will include large scale field tests under more diverse traffic conditions and extend our framework to multi-agent interaction and lateral maneuver decisions. Further, we will study online adaptation and uncertainty-aware learning to further improve robustness and generalization in a more complex real-life environment.

REFERENCES

[1] D. Li and O. Okhrin, "A platform-agnostic deep reinforcement learning framework for effective Sim2Real transfer towards autonomous driving," *Communications Engineering*, vol. 3, no. 1, p. 147, Oct. 2024. [Online]. Available: https://doi.org/10.1038/s44172-024-00292-3

[2] M. Yildirim, B. Dagda, V. Asodia, and S. Fallah, "Behavioral cloning models reality check for autonomous driving," 2024. [Online]. Available: https://arxiv.org/abs/2409.07218

[3] M. Zare, P. M. Kebria, A. Khosravi, and S. Nahavandi, "A survey of imitation learning: Algorithms, recent developments, and challenges," *IEEE Transactions on Cybernetics*, vol. 54, no. 12, pp. 7173–7186, 2024.

[4] Z. Zhang, A. Liniger, D. Dai, F. Yu, and L. Van Gool, "End-to-end urban driving by imitating a reinforcement learning coach," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 15 202–15 212.

[5] Y. Lu, J. Fu, G. Tucker, X. Pan, E. Bronstein, B. Roelofs, B. Sapp, B. White, A. Faust, S. Whiteson, D. Anguelov, and S. Levine, "Imitation is not enough: Robustifying imitation with reinforcement learning for challenging driving scenarios," 2022. [Online]. Available: https://arxiv.org/abs/2212.11419

[6] Z. Huang, J. Wu, and C. Lv, "Efficient deep reinforcement learning with imitative expert priors for autonomous driving," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 10, pp. 7391–7403, 2023.

[7] Z. Li, Y. Wang, H. Wang, Z. Li, P. Li, W. Liu, and Z. Zuo, "From learning to mastery: Achieving safe and efficient real-world autonomous driving with human-in-the-loop reinforcement learning," in *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2025, pp. 7925–7932.

[8] X. Xing and S. Ohl, "Application of a maneuver-based decision making approach for an autonomous system using a learning approach," in *ADVCOMP 2024: The Eighteenth International Conference on Advanced Engineering Computing and Applications in Sciences*, Venice, Italy, September–October 2024, pp. 11–16, sept. 29–Oct. 3, 2024.

[9] X. Xing, M. Molaei, and S. Ohl, "Maneuver-based decision making in autonomous driving via reinforcement learning and simulation-to-real transfer," *International Journal On Advances in Intelligent Systems*, vol. 18, no. 1-2, pp. 46–56, 2025.

[10] "Ki-forum 2025 : Ki in forschung und lehre an hochschulen," H. Homann, C. Rohbani, and J. C. Will, Eds. HannoTalents, 2025, pp. 116–119.

[11] H. Chu, H. Wang, Y. Cheng, A. Li, W. Tian, B. Gao, and H. Chen, "Decision making for autonomous vehicles: A mixed curriculum reinforcement learning approach and a novel safety intervention method," *Transportation Research Part C: Emerging Technologies*, vol. 181, p. 105369, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X25003730

[12] X. Xing and S. Ohl, "Deep reinforcement learning for autonomous driving decision-making in webots," in *2025 11th International Conference on Control, Decision and Information Technologies (CoDIT)*, vol. 1, 2025, pp. 1–6.

[13] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, G. Gordon, D. Dunson, and M. Dudík, Eds., vol. 15. Fort Lauderdale, FL, USA: PMLR, 11–13 Apr 2011, pp. 627–635.

[14] M. Fogli, T. Kudla, B. Musters, G. Pingen, C. Broek, H. Bastiaansen, N. Suri, and S. Webb, "Performance evaluation of kubernetes distributions (k8s, k3s, kubeedge) in an adaptive and federated cloud infrastructure for disadvantaged tactical networks," 05 2021, pp. 1–7.