Adaptive Deep Reinforcement Learning for Critical Boundary Scenario Generation

Junjie Zhou[1,2,4], Lin Wang[1,2], Xiaofan Wang[1,3], Qiang Meng[4]

1. School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai 202400, P. R. China
E-mail: wanglin@sjtu.edu.cn

2. Key Laboratory of System Control and Information Process, Ministry of Education of China, Shanghai 202400, P. R. China

3. School of Electrical and Electronic Engineering, Shanghai Institute of Technology, Shanghai 202400, P. R. China

4. Department of Civil and Environmental Engineering, National University of Singapore, Singapore 117576, Singapore

Abstract: Critical boundary scenarios play a vital role in the comprehensive testing of autonomous vehicles, as they can significantly expedite testing processes and reduce economic and time costs. To tackle the intricate, multidimensional, and diverse nature of intelligent driving scenarios, this paper proposes an innovative bi-level adaptive deep reinforcement learning (BADRL) approach aimed at generating authentic and diverse critical boundary scenarios. Leveraging naturalistic driving data, background agents are trained to exhibit naturalistic and realistic driving behaviors using a neural-based naturalistic driving behavior model. To address the intricacies of multi-interaction, high-dimensional environments, the scenario complexity model is introduced to evaluate the relative complexity between the traffic environment and the tested autonomous vehicle in real time. By integrating the scenario complexity model, the naturalistic driving behavior learning, intelligent driving testing, and critical boundary scenario generation are concatenated together to form a closed loop. BADRL facilitates the upper-level neural network in learning to adaptively increase the complexity of test scenarios, which are then fed into lower-level models to optimize the behavior of principal traffic participants, thereby generating naturalistic and critical boundary test scenarios. Extensive simulations were carried out to verify the efficacy of the BADRL technique in complex intersection environments. Results indicate that the BADRL approach boosts the efficiency of critical boundary scenario generation by approximately 10% compared to the state-of-the-art method.

Key Words: Automated vehicles, Critical Boundary Scenario Generation, Deep Reinforcement Learning, Intelligent Driving Test, Naturalistic Driving Environment

1 Introduction

Due to the rapid progress of autonomous vehicle (AV) technology, expediting the deployment of AVs for real-world applications has become an issue of broad concern around the world [1]. The generation of a vast array of long-tailed critical scenarios is imperative to facilitate comprehensive testing and evaluation of AVs. Research in critical scenario generation is divided into two primary domains based on their intended purposes and applications: collision scenario generation and critical boundary scenario generation [2, 3]. Collision scenario generation entails the development of various perilous situations aimed at provoking collisions with the surrounding environment to assess the safety of AVs. However, the generated collision scenarios only offer insight into the accident rate of AVs, presenting a limited perspective on their comprehensive performance. Furthermore, due to the curse of dimensionality [4], prevailing methods such as importance sampling are effective only in relatively simple, low-dimensional environments [5–7].

Different from collision scenarios, critical boundary scenarios pertain to situations situated at the threshold between safety and collision, serving as a means to expedite the evaluation of the performance boundaries of AVs [8]. Nevertheless, critical boundary scenarios, often intentionally crafted, suffer from a deficit in naturalness, diversity, and generalizability. Thus, the imperative arises to create naturalistic and critical boundary scenarios (NCBS) to facilitate a comprehensive and impartial evaluation of AV performance. Given the pivotal role of NCBS in the advancement of AVs, the generation of NCBS without loss of unbiasedness opens the door to accelerating AV training and comprehensive performance enhancement [1]. Tuncali *et al.* [9] introduced the simulated annealing method to identify critical boundary scenarios linked to lane-changing behavior. Batsch *et al.* [10] proposed the Gaussian process classification method to identify the performance boundary of AVs in traffic jam scenarios. Zhu *et al.* [11] utilized the optimization searching method to search boundary scenarios in the car-following scenario. Wang *et al.* [12] proposed the adaptive sampling method to construct a multi-layer perception-based surrogate model to identify the performance boundary of the intelligent driver model in the car-following scenario. These methods are restricted to identifying existing critical scenarios and are incapable of producing a diverse array of NCBS.

To address these challenges, we propose a bi-level adaptive deep reinforcement learning (BADRL) method to efficiently generate realistic and diverse NCBS in the intricate multi-interaction high-dimensional environment. The basic idea is to develop a neural-based naturalistic driving behavior learning (NNDBL) model and scenario complexity model, and train neural networks to adaptively elevate the test scenario complexity through closed-loop feedback. To reproduce realistic naturalistic traffic scenarios and increase the scenario diversity, we develop an NNDBL model as the background traffic participant's agent to perform complex interaction behaviors with the AV under test. By leverag-

This work was supported by the National Natural Science Foundation of China (No. 62373245, 62336005), the National Key Research and Development Program of China (No. 2023YFB4706800), and in part by the "Dawn" Program of Shanghai Education Commission, China. The author, Prof. Qiang Meng, would like to appreciate the support of the Ministry of Education of Singapore for this study via the research project MOE-000458-00.

ing the scenario complexity model and the automated testing and evaluation framework, we reframe the challenge of NCBS generation into the adaptive enhancement of test scenario complexity. The upper level of the BADRL approach employs neural networks to learn adaptive boosting coefficients of scenario complexity, while the lower layer focuses on the generation of naturalistic and pivotal agent behaviors. Our contributions are manifold and summarized as follows.

1) The BADRL method is designed to transform the NCBS generation problem into adaptive boosting of test scenario complexity. It overcomes the limitation that existing methods are only applicable to low-dimensional scenarios, enhances the naturalness and realism of critical boundary scenario generation, and improves the generation efficiency.
2) Compared to existing trajectory generation methods, our proposed method is more realistic. The proposed NNDBL model can generate diverse and realistic naturalistic traffic scenarios for intelligent driving tests.
3) The proposed method facilitates stochastic scenario generation by merging traffic flow simulation with real-world data. It allows for the bulk creation of critical boundary scenarios that not only align with actual traffic conditions but also adapt to the behavior of the AV under test.
4) We conducted thorough simulations to validate the effectiveness of the BADRL framework. Simulation scenarios are implemented within the Carla simulator, and two distinct self-driving models are employed to demonstrate the efficacy and generalizability of the BADRL method.

The paper is structured as follows: Section 2 elaborates on the BADRL approach, focusing on the naturalistic background agent model, the scenario complexity model, and the bi-level adaptive deep reinforcement learning method. Section 3 outlines the experimental setup and presents the results. Finally, Section 4 provides the conclusion of the paper.

2 Naturalistic and Critical Boundary Scenario Generation

In this section, the BADRL approach is designed to generate NCBS to expedite the evaluation and testing process of AVs from multiple dimensions. The framework of the BADRL approach is illustrated in Fig. 1. The system comprises four key components: NNDBL, scenario complexity model, automated testing and evaluation, and BADRL algorithm. We utilize the encoder-decoder architecture proposed in [13] as the backbone of the NNDBL model to reproduce the behaviors of background traffic participants. Then, we employ naturalistic driving data within realistic traffic environments to conduct unbiased automated testing and assessment for generating NCBS. Moreover, real-world renderings using Carla are employed to enhance the fidelity of the generated NCBS, thereby aligning them more closely with actual traffic scenarios.

2.1 Scenario Complexity Model

To address the complexities of multi-interaction, high-dimensional environments, the scenario complexity model is introduced to assess the relative complexity between the traffic environment and the autonomous vehicle under test in real time. Building upon studies in [14], we define the interacting-pair complexity $C_{i,0}$ as follows:

$$\begin{aligned} C_{i,0} &= \Omega(\theta_{i,0}, d_{i,0}, v_{i,0}) \\ &= f_1(\theta_{i,0}) \times f_2(d_{i,0}) \times f_3(v_{i,0}), \end{aligned} \tag{1}$$

where $f_1(\theta_{i,0})$, $f_2(d_{i,0})$, and $f_3(v_{i,0})$ represent the relationships between the encounter angle $\theta_{i,0}$, relative distance $d_{i,0}$, relative velocity $v_{i,0}$, and the complexity $C_{i,0}$ of the interacting pair, respectively.

The complexities associated with encounter angle, relative distance, and relative velocity can be calculated using the following equations, respectively:

$$f_1(\theta_{i,0}) = \frac{1}{2} - \frac{1}{2}\cos\left(\frac{\theta_{i,0}\pi}{128.57} + \frac{\pi}{15}\right), \theta_{i,0} \in [0, 180), \tag{2}$$

$$f_2(d_{i,0}) = \left(1 - \frac{d_{max} - d_{i,0}}{d_{max} - d_{min}}\right) \times \log\left(\frac{d_{max} - d_{min}}{d_{max} - d_{i,0}}\right), \tag{3}$$

$$f_3(v_{i,0}) = \left(1 - \frac{v_{max} - v_{i,0}}{v_{max} - v_{min}}\right) \times \log\left(\frac{v_{max} - v_{min}}{v_{max} - v_{i,0}}\right), \tag{4}$$

where $d_{i,0}$ represents the relative distance, d_{min} and d_{max} signify the mutual distance in the least and most challenging scenarios. $v_{i,0}$ represents the relative velocity, v_{min} and v_{max} signify the mutual velocity in the least and most challenging scenarios.

The scenario complexity $C(t)$ can be determined by aggregating all interaction pair complexities in the influence area. This can be approximated by summing up the complexities of all interaction pairs.

$$C(t) = \sum_{i=1}^{M} \gamma_i C_{i,0}, \tag{5}$$

where $C(t)$ represents the accumulated scenario complexity, M represents the count of vehicle pairs, γ_i represents the influence of vehicle i, $C_{i,0}$ represents the complexity between vehicle i and AV 0.

2.2 Bi-level Adaptive Deep Reinforcement Learning

To efficiently generate realistic, diverse, transferable, and controllable critical boundary scenarios, we leverage deep reinforcement learning to enhance the behavior of BVs adaptively based on the NNDBL model. Initially, we utilize the NNDBL model to generate a wide array of realistic and diverse naturalistic traffic scenarios. Subsequently, employing the scenario complexity model, we assess the complexity inherent in each test scenario. This problem of generating critical boundary scenarios is then formulated as the sequential Markov decision process. In this context, the actions of BVs are determined by considering both the current state information and the complexity characterization of the scenario. Our objective is to train a DRL policy, implemented as a neural network, capable of directing the maneuvers of BVs. Through adaptive adjustments, this policy aims to elevate the complexity of test scenarios progressively until critical boundary scenarios are effectively generated.

To improve the realism of the generated scenarios as much as possible, we enhance the maneuver of the principal other traffic participants (POTPs) only at critical moments. The identification of critical moments and principal other traffic

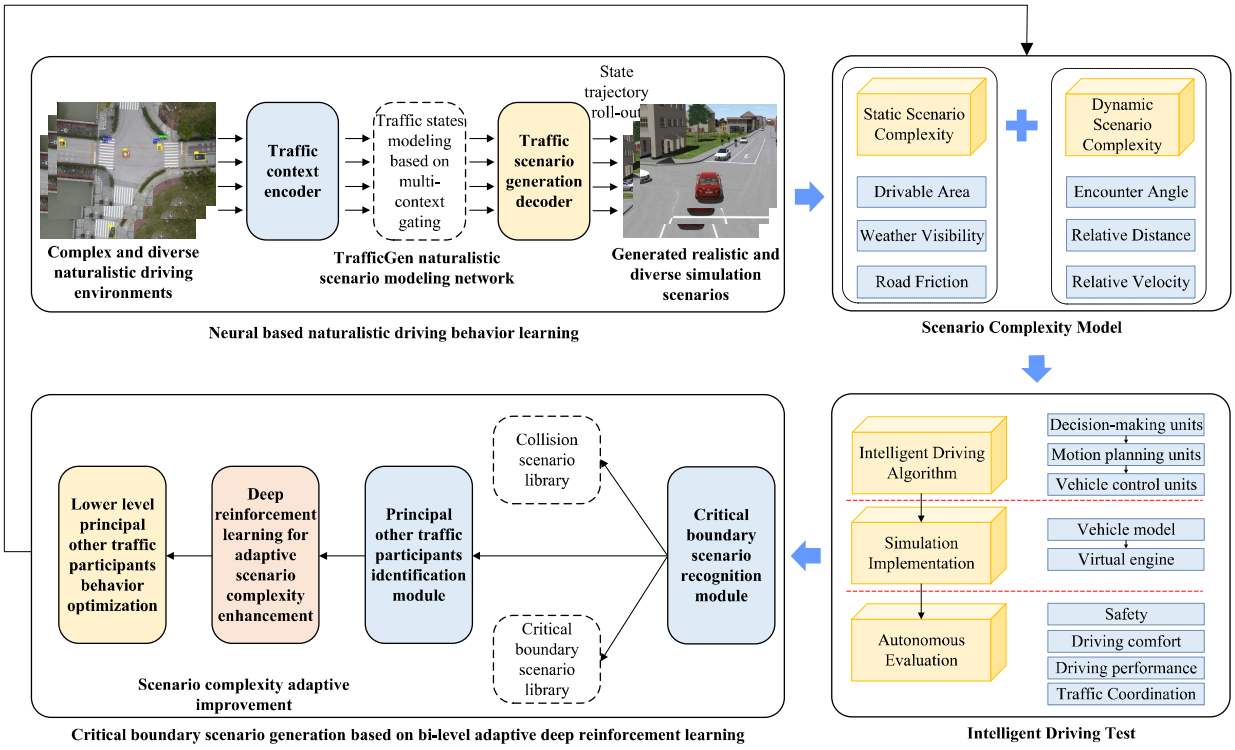

Fig. 1: BADRL System Architecture. The entire system comprises four components, the neural-based naturalistic driving behavior model, the scenario complexity model, the automated testing and evaluation model, and the BADRL algorithm. The actor-critic architecture is proposed to realize realistic, efficient, and diverse critical boundary scenario generation. The actor mainly consists of NNDBL and reinforcement-based adaptive generation of NCBS, while the critic component focuses on automated evaluation of the generated scenarios and AV behaviors.

participants is mainly based on the complexity $C(t)$ defined in the previous section.

After identifying the POTPs, the subsequent challenge lies in generating NCBS rationally and efficiently. The core concept is to systematically enhance POTPs' behavior to amplify the complexity of traffic scenarios. Indeed, the actions of POTPs play a crucial role in shaping the complexity of traffic scenarios, thereby expediting the assessment of AV performance boundaries and the discovery of NCBS [15].

To be applicable to various high-dimensional intelligent driving scenarios and to keep the runtime of the BADRL short, we model the neural network output as the scenario complexity enhancement factor $\beta \in (0, 1]$. This factor enables adaptive optimization of intelligent driving scenarios. Upon successful completion of the current test, the complexity threshold is elevated in preparation for the subsequent test. This threshold adjustment can be determined as follows:

$$C_{crit} = (1 + \beta)C(t). \tag{6}$$

where $C(t)$ denotes the scenario complexity in the naturalistic driving scenario.

The goal of BADRL training is to achieve adaptive scenario complexity boosting to generate realistic and diverse critical boundary scenarios without collisions. Intuitively, we need to train a policy to adaptively increase the complexity of the test scenarios based on the current traffic participant state information and road environment information. To achieve this goal, we derive the reward function in terms of criticality and collision as:

$$\begin{aligned}\mathcal{R} =& r_{crit,t}(x_{AV,t}, x_{POTP,t}, v_{AV,t}, v_{POTP,t}) \\ &+ r_{colli,t} - 2.\end{aligned} \tag{7}$$

where $x_{AV,t}$ and $x_{POTP,t}$ denote the positions of the tested AV and the POTP, $v_{AV,t}$ and $v_{POTP,t}$ denote the velocity of the tested AV and the POTP, respectively. $r_{crit,t}$ serves as a criticality measure, indicating the danger level between the tested AV and the POTPs. The $r_{crit,t}$ consists of two terms, the maximum normalized inverse time-to-collision ($mnTTC^{-1}$) and the minimal post-encroachment time ($mPET$), which can be calculated as:

$$r_{crit,t} = mnTTC^{-1} - \omega \times mPET. \tag{8}$$

where ω denotes the weight parameter used to balance the influence of the two terms.

A larger $mnTTC^{-1}$ implies a more critical interaction behavior, which can be formulated as:

$$mnTTC^{-1} = clip(max\frac{v_i(t) - v_0(t)}{x_i(t) - x_0(t) - L_i}, 0, 2), \tag{9}$$

where L_i denote the length of the POTP i. The clip function employed here serves as a valuable tool in learning networks, setting a threshold for each policy update. This helps stabilize the learning process and prevents many detrimental policy updates.

A smaller $mPET$ means a more critical trajectory conflict, which can be formulated as:

$$mPET = clip(min\frac{PET}{5}, 0, 2), \tag{10}$$

When there is no trajectory conflict between the tested AV and other vehicles, or there are no POTPs, the criticality reward $r_{crit,t}$ is set to 0.

To avoid active collisions between the tested AV and BVs, we introduce $r_{colli,t}$ in Equation (7). In detail, $r_{colli,t}$ can be represented as:

$$r_{colli,t} = \begin{cases} -2, & \text{collision occurred,} \\ 0, & \text{no collision.} \end{cases} \tag{11}$$

After establishing the complexity threshold of the scenario, the second level is to adjust the action u_q to generate NCBS that fulfill the complexity criteria while deviating minimally from the naturalistic driving scenarios (NDS). The trajectories of POTPs obtained from NDS are denoted as $T_{nat}(t, u_{q,nat})$. The enhanced behavioral trajectories are denoted as $T_{crit}(t, u_q)$. We aim to minimize the distribution distance difference between the optimized trajectories and the naturalistic traffic trajectories by optimizing the action of POTPs. The objective function and corresponding constraints of the BADRL method are presented in Equation (12).

$$\begin{aligned} &\arg\min_{u_q} \kappa(u_q), \kappa(u_q) \\ &= \int_{t_0}^{t_0+t_f} (T_{crit}(t, u_q) - T_{nat}(t, u_{q,nat}))^2, \\ &s.t. \quad C(t+1) \geq C_{crit}, \\ &\qquad TTC(t) \geq TTC_{min}, \forall t \in T, \end{aligned} \tag{12}$$

where t_0 represents the initial time of action optimization, t_f represents the interval during which the POTPs influence the behavior of the tested AV, $TTC_i(t)$ signifies the minimum collision time, TTC_{min} represents the minimum safe TTC. The constraints in Equation (12) make sure that no POTPs intentionally collide with the tested AV.

Table 1: The Training Parameters of BADRL

Parameter	Value
Learning rate	0.001
Reward discount factor	0.99
Buffer size	1000000
Batch size	256
Target network update frequency	10000

3 Performance of BADRL Method

To further demonstrate the validity and unbiasedness of the BADRL method, we implemented thorough simulation experiments using data from the real-world intersection, which is intractable for most existing critical boundary scenario generation methods. To validate the effectiveness and generalizability of the generated NCBS, we utilized two different types of self-driving vehicle models for simulation validation. The AV-I model was constructed based on Carla [18] autopilot model, which can be regarded as a black box.

(a) (b)

Fig. 2: The layout of the simulated intersection scenario. (a) A photograph depicting the actual intersection [16]. (b) The simulation intersection reconstructed in Carla to replicate the real-world scenario depicted in (a).

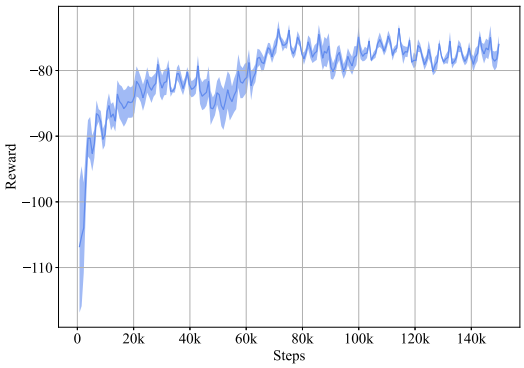

Fig. 3: Learning curve during training of the BADRL method. Rewards are calculated with ten trials, each with four episodes. The shaded area indicates the standard deviation.

The AV-II model was constructed based on the OpenCDA [17] platform, an open-source rule-based autonomous driving algorithm platform that supports multi-vehicle cooperative joint simulation testing.

The simulation intersection scenario is illustrated in Fig. 2. Fig. 2(a) depicts an actual photograph of the intersection from the inD Dataset [16], while Fig. 2(b) showcases the corresponding environment constructed in Carla, faithfully replicating the real scenario depicted in Fig. 2(a).

We implemented the BADRL framework using the deep Q-network (DQN) algorithm [19], a value-based deep reinforcement learning method. DQN employs a target network to derive an unbiased estimator of the mean-squared Bellman error, crucial for training the Q-network. Synchronization between the target network and the Q-network occurs after each iteration period, establishing a coupling between the two networks [20]. The neural network's output is the scenario complexity enhancement factor (β), where the action space is $\beta \in (0, 1]$ with a resolution of 0.2. We utilized PyTorch [21] for training BADRL, leveraging the Adam optimizer [22] to optimize the network parameters. Following meticulous fine-tuning of the parameters, we determined the hyperparameter configurations outlined in Table 1.

The reward curve during DRL training is an important indicator of progress and convergence. The learning curve of the BADRL method during training is shown in Fig. 3. Rewards are calculated with ten trials, each with four episodes.

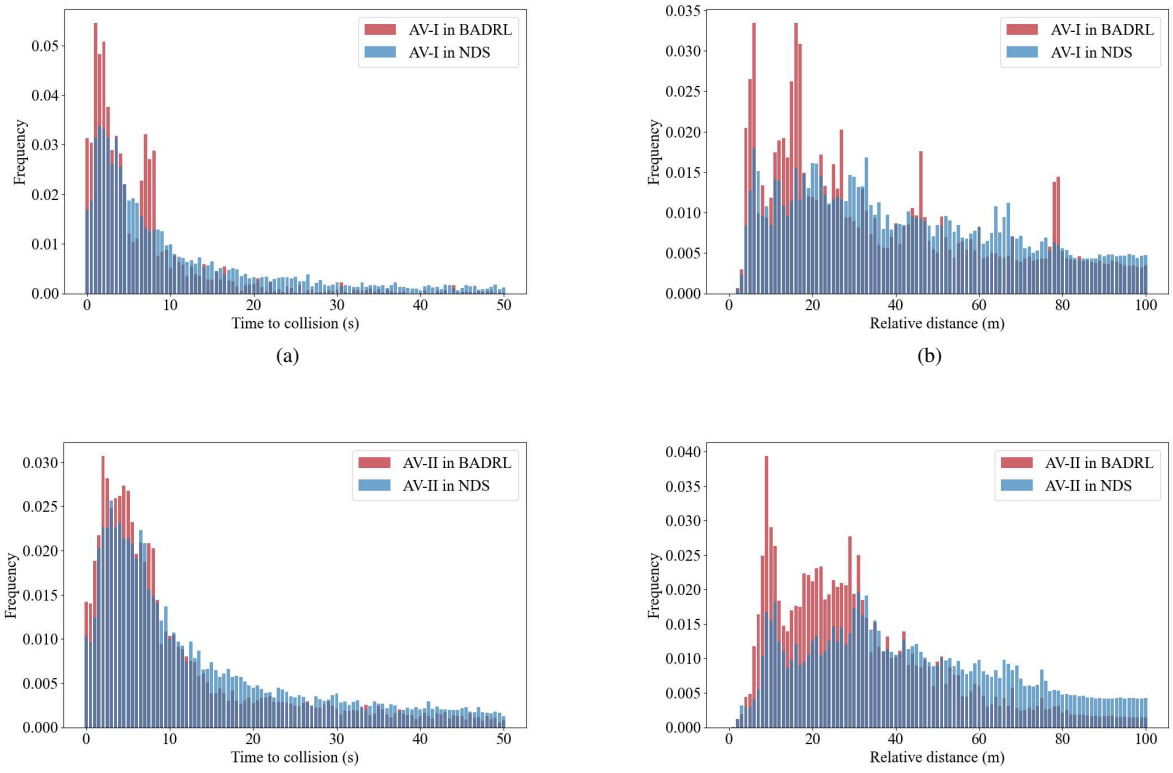

Fig. 4: Performance evaluation of the BADRL-based intelligent testing scenarios. Distribution of time to collision (a) and relative distance (b) for the Carla self-driving model in BADRL and NDS. Distribution of time to collision (c) and relative distance (d) for the OpenCDA self-driving model [17] in BADRL and NDS.

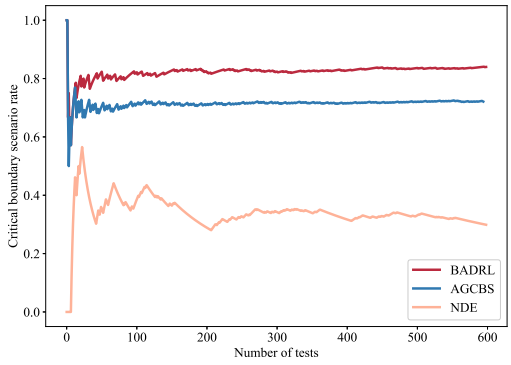

Fig. 5: Comparison of the generation efficiency among the BADRL method, AGCBS method [15], and NDS for critical boundary scenarios. The horizontal axis represents the number of tests, and the vertical axis indicates the proportion of generated critical boundary scenarios relative to the total number of scenarios at each test iteration.

The shaded area indicates the standard deviation. From the figure, it can be seen that the curve tends to stabilize and converge as the training iterates, which reflects the improvement of the BADRL method in NCBS generation.

To demonstrate the effectiveness of the BADRL method, we ran 600 simulation experiments of AVs in both NCBS generated by the BADRL method and NDS, and calculated the time to collision and relative distance distributions between the tested AV and the surrounding vehicles. Fig. 4(a), (b) show the distribution of time to collision and relative distance for Carla autonomous driving model in NDS and BADRL. From the figures, it can be seen that the NCBS generated by BADRL are more hazardous and critical with shorter time to collision and relative distance compared to the NDS. In addition, it can be found that the distribution between BADRL and NDS is highly consistent. This proves that the BADRL method can effectively and unbiasedly generate various boundary scenarios. Fig. 4(c), (d) show the distribution of time to collision and relative distance for the OpenCDA autonomous driving model in NDS and BADRL. As can be seen from Fig. 4, the frequency distribution of the OpenCDA autonomous driving model is notably smaller than that of the Carla autonomous driving model within intervals where the time to collision is less than ten. Specifically, the peak frequency measures only 0.03, contrasting with the 0.05 peak frequency observed in the Carla autonomous driving model. Furthermore, the OpenCDA autonomous driving model exhibits a larger relative distance distribution. These findings provide evidence supporting the assertion that the OpenCDA autonomous driving model offers enhanced safety compared to the Carla autonomous driving model.

The comparison of the generation efficiency among the BADRL method, AGCBS method [15], and NDS for critical boundary scenarios is depicted in Fig. 5. The horizon-

tal axis represents the number of tests, and the vertical axis indicates the ratio of generated critical boundary scenarios to the total scenarios for the current number of tests. Fig. 5 demonstrates that the proposed BADRL method enhances the efficiency of critical boundary scenario generation by approximately 10% compared to the state-of-the-art AGCBS method.

4 Conclusion

To address the significant challenges stemming from the economic and temporal costs associated with the comprehensive testing and validation of AVs, we propose the BADRL approach aimed at generating realistic and diverse critical boundary scenarios. Our methodology leverages naturalistic driving data to train background agents using a neural-based naturalistic driving behavior model. Furthermore, we introduced a scenario complexity model to adaptively adjust the complexity of test scenarios in real-time. The BADRL approach facilitates real-time adaptive enhancement of scenario complexity, enabling the generation of compelling NCBS in high-dimensional complex environments.

Extensive simulations were conducted in complex intersection environments to validate the effectiveness of the BADRL approach using the Carla simulation. The results demonstrate that the proposed BADRL method enhances the efficiency of critical boundary scenario generation by approximately 10% compared to the state-of-the-art methods. The simulation results indicate that our method has the potential to address the current limitations in testing and validation processes for AVs, paving the way for accelerated testing and application of AVs in the future.

References

[1] S. Feng, H. Sun, X. Yan, H. Zhu, Z. Zou, S. Shen, and H. X. Liu, "Dense reinforcement learning for safety validation of autonomous vehicles," *Nature*, vol. 615, no. 7953, pp. 620–627, 2023.

[2] J. Sun, H. Zhang, H. Zhou, R. Yu, and Y. Tian, "Scenario-based test automation for highly automated vehicles: A review and paving the way for systematic safety assurance," *IEEE transactions on intelligent transportation systems*, vol. 23, no. 9, pp. 14 088–14 103, 2021.

[3] W. Ding, C. Xu, M. Arief, H. Lin, B. Li, and D. Zhao, "A survey on safety-critical driving scenario generation—a methodological perspective," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 7, pp. 6971–6988, 2023.

[4] S. Feng, Y. Feng, H. Sun, S. Bao, Y. Zhang, and H. X. Liu, "Testing scenario library generation for connected and automated vehicles, part II: Case studies," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 9, pp. 5635–5647, 2021.

[5] D. Zhao, H. Lam, H. Peng, S. Bao, D. J. LeBlanc, K. Nobukawa, and C. S. Pan, "Accelerated evaluation of automated vehicles safety in lane-change scenarios based on importance sampling techniques," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 3, pp. 595–607, 2017.

[6] D. Zhao, X. Huang, H. Peng, H. Lam, and D. J. LeBlanc, "Accelerated evaluation of automated vehicles in car-following maneuvers," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 3, pp. 733–744, 2018.

[7] S. Kuutti, S. Fallah, and R. Bowden, "Training adversarial agents to exploit weaknesses in deep control policies," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 108–114.

[8] J. Zhou, L. Wang, and X. Wang, "Scalable evaluation methods for autonomous vehicles," *Expert Systems with Applications*, vol. 249, p. 123603, 2024.

[9] C. E. Tuncali, T. P. Pavlic, and G. Fainekos, "Utilizing s-taliro as an automatic test generation framework for autonomous vehicles," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, 2016, pp. 1470–1475.

[10] F. Batsch, A. Daneshkhah, M. Cheah, S. Kanarachos, and A. Baxendale, "Performance boundary identification for the evaluation of automated vehicles using gaussian process classification," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 419–424.

[11] B. Zhu, P. Zhang, J. Zhao, and W. Deng, "Hazardous scenario enhanced generation for automated vehicle testing based on optimization searching method," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 7321–7331, 2022.

[12] Y. Wang, R. Yu, S. Qiu, J. Sun, and H. Farah, "Safety performance boundary identification of highly automated vehicles: A surrogate model-based gradient descent searching approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 23 809–23 820, 2022.

[13] L. Feng, Q. Li, Z. Peng, S. Tan, and B. Zhou, "Trafficgen: Learning to generate diverse and realistic traffic scenarios," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 3567–3575.

[14] C. Tam and R. Bucknall, "Cooperative path planning algorithm for marine surface vessels," *Ocean Engineering*, vol. 57, pp. 25–33, 2013.

[15] J. Zhou, L. Wang, and X. Wang, "Online adaptive generation of critical boundary scenarios for evaluation of autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 6, pp. 6372–6388, 2023.

[16] J. Bock, R. Krajewski, T. Moers, S. Runde, L. Vater, and L. Eckstein, "The ind dataset: A drone dataset of naturalistic road user trajectories at german intersections," in *2020 IEEE Intelligent Vehicles Symposium (IV)*, 2020, pp. 1929–1934.

[17] R. Xu, Y. Guo, X. Han, X. Xia, H. Xiang, and J. Ma, "Opencda: An open cooperative driving automation framework integrated with co-simulation," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, 2021, pp. 1155–1162.

[18] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "Carla: An open urban driving simulator," in *Conference on Robot Learning (CoRL)*, 2017, pp. 1–16.

[19] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[20] J. Fan, Z. Wang, Y. Xie, and Z. Yang, "A theoretical analysis of deep q-learning," in *Learning for Dynamics and Control (L4DC)*, 2020, pp. 486–489.

[21] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[22] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.