Robustness Verification of a Reinforcement Learning-based Agent for Automated Car Parking

Francesco Bellotti[1], Riccardo Berta[1], Vafali Soltanmuradov[1],
David Martín Gómez[2], Akshay Dhonthi[3], Vahid Hashemi[3], and Luca Lazzaroni[1]

[1] Department of Electrical, Electronic and Telecommunication Engineering (DITEN), University of Genoa, Via Opera Pia 11a, 16145 Genova, Italy
[2] Department of Electrical Engineering, Universidad Carlos III de Madrid, 28911 Leganes, Spain
[3] AUDI AG, Auto-Union-Straße 1, 85057 Ingolstadt, Germany
vafali.soltanmuradov@edu.unige.it

Abstract. Safety critical systems require careful verification of their performance in well-defined operational conditions. We have verified the performance of an automated driving (AD) parking agent trained through a reinforcement learning-based automated driving (AD) parking agent in three exemplary critical scenarios in the CARLA driving simulation environment. Results, obtained through the Marabou formal verification tool, show that the system does not produce outputs in a range that would violate expert-defined safety assumptions. Analyzing robustness to input signal perturbation, we observe that injection of Gaussian and pixel noise decreases slightly the safety performance. Interestingly, especially at higher noise levels, the agent frequently decides to remain stationary, which would allow the system to safely issue a take-over request to the driver. These results indicate significant further research in the field.

Keywords: Verification, Marabou, automated parking, noise, lidar, reinforcement learning.

1 Introduction

Automated parking is an automated driving (AD) function that is now spreading also at the commercial level [1]. However, research in the field is very active [2]. Adoption of AD functions (ADFs) requires robustness. To this end, various techniques and tools have been developed for verification and testing (e.g., [3]). This is particularly important for neural network (NN) architectures, whose decision-making process is substantially a black box, that needs very careful handling (e.g., [4]).

Reinforcement learning (RL) has emerged as a powerful tool for training decision-making agents for AD functions (e.g., [5, 6]), particularly benefiting from the latest advancements in simulation technologies (e.g., [7]). RL agents learn optimal policies through interactions with the environment, by trial and error. However, robustness of such RL agents, especially in safety-critical applications, remains a concern [8]. Ensuring that these agents can handle a wide range of scenarios without failure and in an explainable way (e.g., [9]) is crucial for their deployment in real-world settings.

This study investigates effectiveness of robustness verification of a deep RL (DRL) agent trained for automated parking. We apply Marabou [10], a state-of-the-art open-source framework for NN verification and analysis, to a DRL automated parking agent [11] we developed in the Hi-Drive project [12]. In the experiments, we define a set of parking maneuver scenarios within the CARLA driving simulator [13], with noise. The goal is to propose a benchmark for robustness automated parking verification.

2 Related Work

Verification of NNs is crucial in ensuring that automated parking systems operate reliably and safely under various conditions. Verification is a complex task due to the non-linear and high-dimensional nature of NNs. The goal is to provide guarantees about network behavior under specific conditions, such as the robustness against adversarial attacks or the absence of undesirable outputs. Verification methods include Satisfiability Modulo Theories (SMT) solving [14], Mixed Integer Linear Programming (MILP) [15], and Reachability Analysis [16].

Some open-source tools have been developed to support research on automated verification. VeriNet employs MILP for verification and a symbolic interval propagation-based approach for formal guarantees of safety, security, correctness, and robustness [17]. VeriNetBF is an extension to verify NN image classifiers against intensity perturbations in computer vision, also addressing scalability [18]. Marabou is another state-of-the-art tool for verifying deep NNs [19]. It encodes the network and property to be verified into constraints, that are solved using a combination of linear programming (LP) and SMT. Marabou supports different NN activation functions, topologies, parallel execution, and multiple input formats. Grese et al. [20] present a detailed use case for local robustness verification and sensitivity analysis. Liu et al. [21] verify an aircraft collision avoidance NN trained through RL. Besides studying local robustness, the authors also analyze six schematic, critical scenarios (implemented in Marabou as sets of input constraints), verifying, with mixed results, whether the NN's single output (angular speed) could violate the expectations in those contexts. Taking inspiration from this, we transfer the analysis to the AD domain, utilizing a more complex simulation environment and sensor and actuator architecture, and also measuring the impact of various types and intensities of noise on the agent's behavior.

3 Experiment

We use Marabou to verify the decision-making of the mentioned DRL agent for real-time trajectory planning and tracking [11], with different perturbation levels. Marabou takes in input a set of constraints on the agent observations and actions (i.e., its NN inputs and output, respectively) and verifies their satisfaction. This is used to exclude that some undesirable actions are performed by the agent in the scenarios specified by the observation constraints. The agent, which achieves a 96% success rate in parking a car in a target lot (e.g., Fig. 1), controls a CARLA-simulated Audi e-tron car, equipped with a lidar sensor placed at the center of the roof, having a 360° horizontal field of view. The model takes in input 61 complanar lidar rays and some vehicular signals, and

outputs throttle, brake, steer (the maximum steering angle is 35°, with NVIDIA's PhysX model [22]), and reverse commands, all normalized between [-1, 1]. 5 input frames are stacked at each decision point (5 Hz frequency), to make the model aware of the car's latest dynamic (1 second). We focus on three potential collision scenarios (Fig. 1). Distances and angles were chosen arbitrarily, but with the goal of defining challenging settings, where the agent should avoid dangerous actions, even in the presence of noise. The three data points were taken from three successful simulation episodes of the agent.

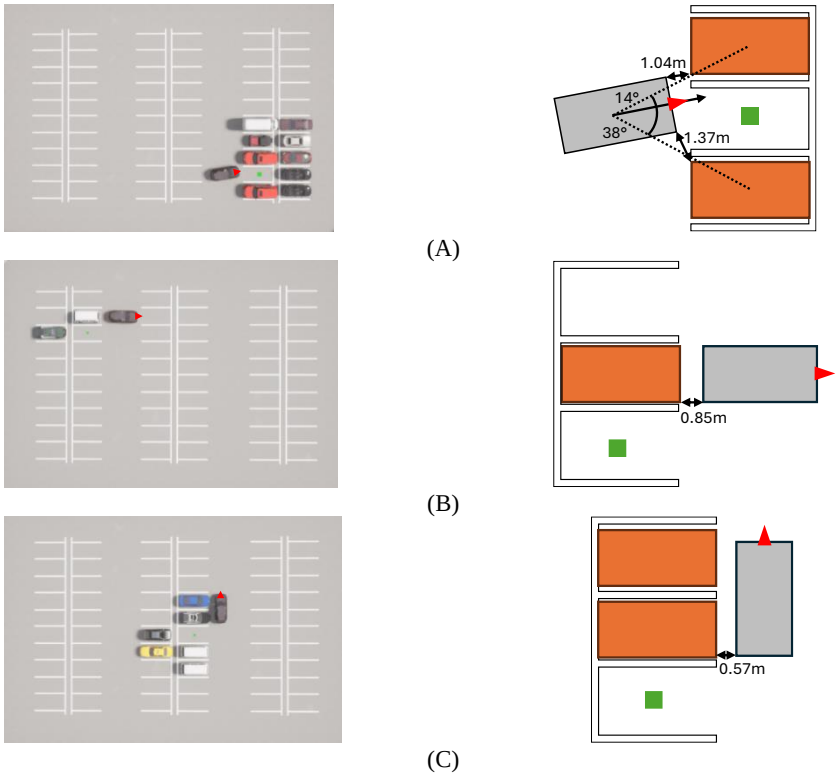

Fig. 1. Tested critical scenarios. The front of the ego vehicle is indicated with a red triangle. In (A) the agent has to avoid the two vehicles positioned on either side of the parking slot, in (B) the vehicle behind, and in (C) the two vehicles on its left.

Study of perturbations is important to understand how robust an NN is to variations and inaccuracies in real-world data, which is not immediate to assess, given the opaque nature of an NN's decision process. By examining how perturbations impact the NN, we can identify potential weaknesses and consequently improve the model's robustness.

To study perturbations in the lidar signal, we implemented Gaussian noise [23], a statistical noise with a probability density function equal to that of the normal distribution. We add it to the original signal, as it is often used to simulate real-world inaccuracies and disturbances in signals [24]. We also implement pixel noise, aka "salt and pepper" in image processing. It completely alters the value of one or more pixels, depending on its intensity, while staying in the [-1, 1] range (differently from the

additive Gaussian noise, which may exceed the bounds). Pixel noise may simulate a permanent pixel failure or rain, since raindrops may interfere with laser pulses, causing reflections and distortions, which may result in false detections or missed objects.

The first experiment phase involved testing the three critical scenarios using clean (i.e., noiseless) input. For each scenario, we defined two exemplary sets of output constraints (Table 1). We defined each constraint as a set of output value ranges that, if satisfied, could represent a collision-risk. The values were defined empirically based on trials in the simulator. For instance, the two constraints for scenario A check that the vehicle does not take a strong acceleration to the left (A1 constraint) or the right (A2). In the second scenario we check that the agent does not proceed in reverse either with middle-low (B1) or high throttle levels (B2). The third scenario constraints check that the vehicle does not take a strong left (C1) nor right (C2) turn.

Then, we introduced Gaussian noise to the lidar sensor signals, starting with a low level of noise and gradually increasing it. Finally, we did the same for pixel noise. For each test involving noise, we run 1,000 iterations, to account for the environment's stochasticity, counting the number of unsatisfactory (UNSAT) and satisfactory (SAT) verification outcomes. Marabou returns SAT if an input exists for which the NN satisfies the given property or set of constraints. For each SAT outcome, Marabou also provides a counterexample that fulfills the specified conditions. If Marabou returns UNSAT, it indicates that no input can satisfy the given property or set of constraints.

4 Results

Verification of all test scenarios using clean (i.e., noiseless) input is summarized in Table 1, which shows that, in the absence of noise, the NN always fulfills the stated expectations (i.e., the reported set of outcomes is never produced by the agent).

Table 1. Verification of parking scenarios in no-noise conditions.

Scenario	Id	Constraints	Result
A	A1	Throttle ≥ 0.9, Steering ≤ -0.75, Brake ≤ 0.01, Reverse = False	UNSAT
	A2	Throttle ≥ 0.9, Steering ≥ 0.75, Brake ≤ 0.01, Reverse = False	UNSAT
B	B1	0.05 ≤ Throttle ≤ 0.5, Brake ≤ 0.01, Reverse = True	UNSAT
	B2	Throttle ≥ 0.75, Brake ≤ 0.01, Reverse = True	UNSAT
C	C1	Throttle ≥ 0.25, Steering ≤ -0.75, Brake ≤ 0.01, Reverse = False	UNSAT
	C2	Throttle ≥ 0.25, Steering ≥ 0.75, Brake ≤ 0.01, Reverse = True	UNSAT

Results (reported scenario-wise for Gaussian noise, and pixel noise in Fig. 2, 3, and 4) show that, in almost all cases (e.g., A1 Gaussian and A1 Pixel), there is an initial decrease in UNSAT rates. This was expected, as it indicates that the agent is being increasingly misled by the increasingly noisy signal measurements. However, once a threshold is overcome (which varies with scenarios and constraints, e.g., 0.6 noise intensity in A1 Gaussian or 0.4 attack ratio in A1 Pixel), the trend is inverted. This may be understood through observation. In several cases (their percentage is represented in dark blue in the bar charts), the agent decides to remain stationary rather than taking risky actions (even if it was not trained in such noisy conditions). After a certain threshold (e.g., 0.6 noise intensity in A1 Gaussian or 0.2 attack ratio in A1 Pixel), as

noise significantly increases, the agent understands the difficulty and correspondingly increases the percentage of times it avoids moving, thus reducing the percentage of constraint violation. On the other hand, results show that there is a percentage of cases (particularly at low-mid noise levels) in which the agent is deceived (i.e., a constraint is SAT, indicating violation of safe behavior, as it happens in 11% of the cases in A1 Gaussian). In several other perturbation cases, the agent remains stationary in a safe condition, in which the vehicular system may issue a take-over request to the driver [25] or an alternative maneuver. The only one exception to the trend inversion pattern is represented by the second constraint in (A2). In this case, the vehicle is more robust towards a strong right acceleration, which deserves more in-depth analysis, also considering the agent's actual decision in the recorded case.

Considering the noise types, we observe that Gaussian is more impactful, leading to a greater reduction in UNSAT rates. Differently from the Gaussian case, we also notice a saturation, in pixel noise, of the stationary case frequency to 50% (it never overcomes 60%). With Gaussian additive noise, as standard deviation becomes greater than 1, the input more frequently exceeds the [-1, 1] training range, and correspondingly the vehicle shows a greater tendency to stand still. Notably, the agent was not trained for this behavior with noise. But we argue that, through conservative training (involving significant collision penalties), the agent learned to move only in safe cases. This is significant, also considering that the agent achieves an overall 97% goal reach rate [11].

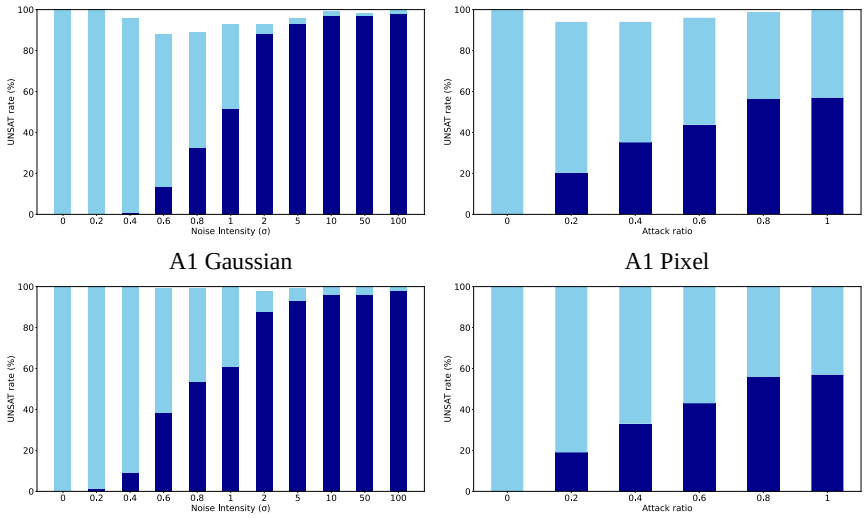

Fig. 2. UNSAT rates for scenario (A), with different constraints (1, 2, cf. Table 1) and types of noise. Dark blue fill indicates the fraction of UNSAT cases in which the vehicle remains stationary, while light blue color denotes the fraction in which the vehicle moves.

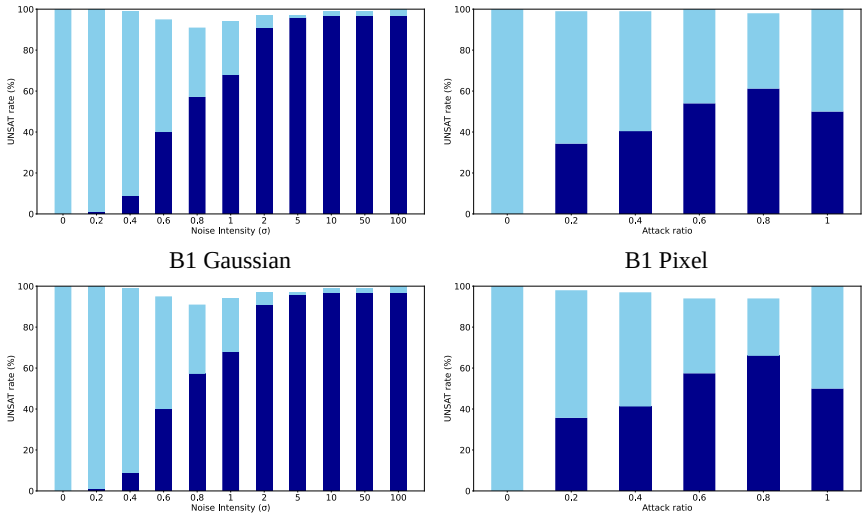

Fig. 3. Same as Fig. 2, for scenario (B).

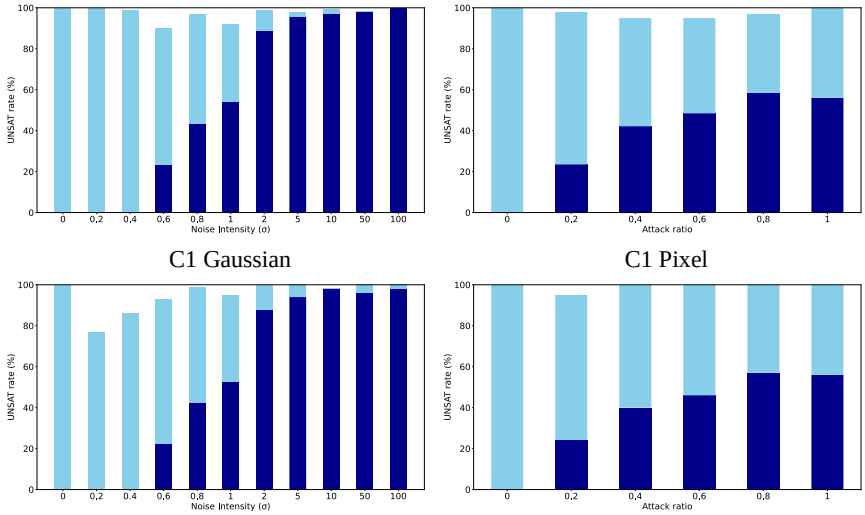

Fig. 4. Same as Fig. 2, for scenario (C).

5 Conclusions and Future Work

Safety critical systems require careful verification in well-defined operational conditions. We have verified performance of a DRL-based AD parking agent in three critical scenarios. Results show that the system does not produce outputs in a range that

would violate expert-defined safety assumptions. Analyzing robustness to input signal perturbation, we observe that injection of Gaussian and pixel noise decreases slightly the safety performance. Interestingly, especially at higher noise levels, the agent frequently decides to remain stationary, which would allow the system to issue a take-over request to the driver. To the best of our knowledge, this is the first analysis applying an automated verification tool in AD parking critical scenarios.

Future work should consider input range constraint verification. Other research direction may include local robustness, sensitivity and liveness analysis.

Acknowledgments

The authors thank the Hi-Drive partners, particularly Sabine Rieder for her contribution. This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006664. The sole responsibility of this publication lies with the authors. Neither the European Commission nor CINEA – in its capacity of Granting Authority – can be made responsible for any use that may be made of the information this document contains.

References

1. de Visser, E.J., Phillips, E., Tenhundfeld, N., Donadio, B., Barentine, C., Kim, B., Madison, A., Ries, A., Tossell, C.C.: Trust in automated parking systems: A mixed methods evaluation. Transp. Res. Part F Traffic Psychol. Behav. 96, 185–199 (2023). https://doi.org/10.1016/j.trf.2023.05.018.
2. Jo, Y., Ha, J., Hwang, S.: Survey of Technology in Autonomous Valet Parking System. Int. J. Automot. Technol. 24, 1577–1587 (2023). https://doi.org/10.1007/s12239-023-0127-1.
3. Wu, M., Wu, H., Barrett, C.: VeriX: Towards Verified Explainability of Deep Neural Networks, http://arxiv.org/abs/2212.01051, (2023). https://doi.org/10.48550/arXiv.2212.01051.
4. Haar, L.V., Elvira, T., Ochoa, O.: An analysis of explainability methods for convolutional neural networks. Eng. Appl. Artif. Intell. 117, 105606 (2023). https://doi.org/10.1016/j.engappai.2022.105606.
5. Gu, Z., Gao, L., Ma, H., Li, S.E., Zheng, S., Jing, W., Chen, J.: Safe-State Enhancement Method for Autonomous Driving via Direct Hierarchical Reinforcement Learning. IEEE Trans. Intell. Transp. Syst. 24, 9966–9983 (2023). https://doi.org/10.1109/TITS.2023.3271642.
6. Berta, R., Lazzaroni, L., Capello, A., Cossu, M., Forneris, L., Pighetti, A., Bellotti, F.: Development of deep-learning-based autonomous agents for low-speed maneuvering in Unity. J. Intell. Connect. Veh. (in press). https://doi.org/10.26599/JICV.2023.9210039.
7. Li, H., Han, R., Zhao, Z., Xu, W., Hao, Q., Wang, S., Xu, C.: Seamless Virtual Reality With Integrated Synchronizer and Synthesizer for Autonomous Driving. IEEE Robot. Autom. Lett. 9, 4218–4225 (2024). https://doi.org/10.1109/LRA.2024.3375266.
8. Amir, G., Corsi, D., Yerushalmi, R., Marzari, L., Harel, D., Farinelli, A., Katz, G.: Verifying Learning-Based Robotic Navigation Systems, https://doi.org/10.48550/arXiv.2205.13536.
9. Bellotti, F., Lazzaroni, L., Capello, A., Cossu, M., De Gloria, A., Berta, R.: Explaining a Deep Reinforcement Learning (DRL)-Based Automated Driving Agent in Highway Simulations. IEEE Access. 11, 28522–28550 (2023). https://doi.org/10.1109/ACCESS.2023.3259544.

10. Katz, G., Huang, D.A., Ibeling, D., Julian, K., Lazarus, C., Lim, R., Shah, P., Thakoor, S., Wu, H., Zeljić, A., Dill, D.L., Kochenderfer, M.J., Barrett, C.: The Marabou Framework for Verification and Analysis of Deep Neural Networks. In: Dillig, I. and Tasiran, S. (eds.) Computer Aided Verification. pp. 443–452. Springer International Publishing (2019). https://doi.org/10.1007/978-3-030-25540-4_26.
11. Lazzaroni, L., Pighetti, A., Bellotti, F., Capello, A., Cossu, M., Berta, R.: Automated Parking in CARLA: A Deep Reinforcement Learning-Based Approach. In: Bellotti, F., Grammatikakis, M.D., Mansour, A., Ruo Roch, M., Seepold, R., Solanas, A., and Berta, R. (eds.) Applications in Electronics Pervading Industry, Environment and Society. pp. 352–357. Springer Nature (2024). https://doi.org/10.1007/978-3-031-48121-5_50.
12. Capello, A., Fresta, M., Bellotti, F., Haghighi, H., Hiller, J., Mozaffari, S., Berta, R.: Exploiting Big Data for Experiment Reporting: The Hi-Drive Collaborative Research Project Case. Sensors. 23, 7866 (2023). https://doi.org/10.3390/s23187866.
13. Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., Koltun, V.: CARLA: An Open Urban Driving Simulator, http://arxiv.org/abs/1711.03938, (2017). https://doi.org/10.48550/arXiv.1711.03938.
14. Amir, G., Wu, H., Barrett, C., Katz, G.: An SMT-Based Approach for Verifying Binarized Neural Networks. In: Groote, J.F. and Larsen, K.G. (eds.) Tools and Algorithms for the Construction and Analysis of Systems. pp. 203–222. Springer International Publishing (2021). https://doi.org/10.1007/978-3-030-72013-1_11.
15. Wang, H., Liu, J., Chen, X., Wang, X., Li, P., Yin, W.: DIG-MILP: a Deep Instance Generator for Mixed-Integer Linear Programming with Feasibility Guarantee, http://arxiv.org/abs/2310.13261, (2023). https://doi.org/10.48550/arXiv.2310.13261.
16. Lew, T., Pavone, M.: Sampling-based Reachability Analysis: A Random Set Theory Approach with Adversarial Sampling, http://arxiv.org/abs/2008.10180, (2020).
17. Henriksen, P., Lomuscio, A.: Efficient Neural Network Verification via Adaptive Refinement and Adversarial Search. In: ECAI 2020. pp. 2513–2520. IOS Press (2020). https://doi.org/10.3233/FAIA200385.
18. Henriksen, P., Hammernik, K., Rueckert, D., Lomuscio, A.: Bias Field Robustness Verification of Large Neural Image Classifiers. In: BMVC. p. 202 (2021).
19. Wu, H., Isac, O., Zeljić, A., Tagomori, T., Daggitt, M., Kokke, W., Refaeli, I., Amir, G., Julian, K., Bassan, S., Huang, P., Lahav, O., Wu, M., Zhang, M., Komendantskaya, E., Katz, G., Barrett, C.: Marabou 2.0: A Versatile Formal Analyzer of Neural Networks, http://arxiv.org/abs/2401.14461, (2024).
20. Grese, J.M., Pasareanu, C., Pakdamanian, E.: Formal Analysis of a Neural Network Predictor inShared-Control Autonomous Driving. In: AIAA Scitech 2021 Forum. American Institute of Aeronautics and Astronautics. https://doi.org/10.2514/6.2021-1580.
21. Liu, C., Cofer, D., Osipychev, D.: Verifying an Aircraft Collision Avoidance Neural Network with Marabou. In: Rozier, K.Y. and Chaudhuri, S. (eds.) NASA Formal Methods. pp. 79–85. Springer Nature (2023). https://doi.org/10.1007/978-3-031-33170-1_5.
22. Welcome to PhysX — NVIDIA PhysX SDK 3.4.0 Documentation, https://docs.nvidia.com/gameworks/content/gameworkslibrary/physx/guide/Manual/Introduction.html, last accessed 2024/09/12.
23. Tsiligkaridis, T., Tsiligkaridis, A.: Diverse Gaussian Noise Consistency Regularization for Robustness and Uncertainty Calibration, http://arxiv.org/abs/2104.01231, (2023). https://doi.org/10.48550/arXiv.2104.01231.
24. Jang, G., Lee, W., Son, S., Lee, K.M.: C2N: Practical Generative Noise Modeling for Real-World Denoising, http://arxiv.org/abs/2202.09533, (2022). https://doi.org/10.48550/arXiv.2202.09533.
25. Pipkorn, L., Tivesten, E., Flannagan, C., Dozza, M.: Driver Response to Take-Over Requests in Real Traffic. IEEE Trans. Hum.-Mach. Syst. 53, 823–833 (2023). https://doi.org/10.1109/THMS.2023.3304003.