Synthetic Dataset Generation Using Logical Scenario Files for Automotive Perception Testing

Mikel García[1,2], Aitor Iglesias[1,2], Martí Sánchez[1,2], Ruben Naranjo[1,2], Jon Ander Iñiguez de Gordoa[1,2], Marcos Nieto[1] and Naiara Aginako[2]

Abstract— **Conducting extensive recording campaigns to asses the safety of newly developed Automated Driving Systems (ADS) or perception algorithms has been proved to be a costly and time consuming process. This is one of the reasons why the automotive industry is adopting the scenario-based testing methodology, to verify and validate the safety of the developed ADS in their expected operating domain. The exterior perception system is the first component in the sense-plan-act process of Connected Cooperative and Automated Vehicles (CCAVs). In this context, high-fidelity simulation engines are used to replicate sensor setups at reduced cost and higher scalability than driving and capturing data from real sensors. The use of logical automotive scenario descriptions allows defining certain parameter ranges, contexts and actions to execute simulations that fulfill the desired conditions. This work proposes a methodology for generating synthetic labelled datasets to test and validate automotive perception systems using logical scenario files. Decoupling the desired sensor setup from the simulation allows reproduction and testing the same situation under different sensor setups and conditions. We implement the methodology to validate three 3D LiDAR-based object detectors in three different sensor setups. The generated sample dataset will be made public here[1].**

I. Introduction

The automotive sector is undergoing constant change to achieve autonomous driving. Nowadays, vehicles rely on a wide variety of sensors, actuators, algorithms, and machine learning systems that operate in real time for a multitude of automated driving systems (ADS). One of the key components of modern vehicles is their perception system. The perception system of a vehicle is generally composed of a combination of sensors, including cameras, LiDARs, or radars. This system aids the vehicle in understanding its local environment, and its proper functioning is critical for the dynamic driving task (DDT) of the ego-vehicle and the future of Connected Cooperative and Automated Vehicles (CCAVs) and C-ITS Day2+ services, where perception system data can be transmitted via Vehicle-to-Everything (V2X) using ETSI CPM [1].

Traditionally, real world recording campaigns have been conducted to create datasets to test the functionality of various autonomous driving systems or to train machine learning models, including perception systems. This has led to multiple open-source datasets with terabytes of automotive labelled data from different sensor types and setups, some of the most popular datasets include NuScenes [2], Waymo [3] or KITTI [4], that can be considered a pioneer dataset in the field. However, as demonstrated in [5], hundreds of millions or even billions of kilometers are needed to prove that an autonomous driving function performs better than humans in terms of fatalities and injuries, which is almost impossible to achieve due to the time and cost limitations of performing such a recording campaign. Nonetheless, even if that amount of data is recorded, most of the captured data would be repetitive (i.e., straight driving on a highway) and would need to be processed to extract relevant situations. As a result, the concept of scenario-based testing (SBT) proposed in ISO26262 [6] and the recently published ISO 21448 [7] that focuses on the Safety of the Intended Functionality (SOTIF) is the currently used methodology in the automotive field to prove the safety of ADS. This approach aims to reduce the required amount of data by testing an ADS under relevant scenarios. Even using SBT, reproducing certain situations in a real environment may be infeasible, not only for the implicit risk in reproducing certain maneouvres, but also for the high cost involved in performing such tests. These are the main reasons why usually ADS validation is combined with virtual testing using high-fidelity simulation engines, as can be seen in the multiple synthetic datasets replicating real vehicle sensor setups [8]–[11]. This allows a complementary testing and validation phase at a reduced cost and better scalability.

During this work, when we use the term *scenario*, it should be understood as the definition provided in [12]: ”*A scenario is a quantitative description of the relevant characteristics and activities and/or goals of the ego vehicle(s), the static environment, the dynamic environment, and all events that are relevant to the ego vehicle(s) within the time interval between the first and the last relevant event*”. Scenarios have been categorised into different types depending on their abstraction level. In the PEGASUS[2] project, three levels of abstraction were introduced [13], classifying scenarios into

The work of this paper has been done in the context of the SUNRISE project which is co-funded by the European Commission's Horizon Europe Research and Innovation Programme under grant agreement number 101069573. Views and opinions expressed, are those of the author(s) only and do not necessarily reflect those of the European Union or the European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them.

[1] Vicomtech Foundation, Basque Research and Technology Alliance (BRTA), Mikeletegi 57, 20009, Donostia-San Sebastián (Spain).

[2] University of the Basque Country (UPV/EHU), Donostia-San Sebastián (Spain).

[1]The link to the dataset will be included in the camera-ready version of the paper.

[2]`www.pegasusprojekt.de/en`

Manuscript 179 submitted to 2025 36th IEEE Intelligent Vehicles Symposium (IV). Received January 30, 2025.

functional, logical and concrete scenarios. Functional scenarios would correspond to a human-readable description using natural language of a given scenario; logical scenarios would include parameter ranges and relations between the different entities of a given scenario under a descriptive format; and concrete scenarios would be derived from logical scenarios by assigning fixed values to the defined the parameter ranges. In [14] an additional type of scenario is introduced, by adding abstract scenario types as a formalised machine-readable description of a scenario that would correspond to the second level of abstraction after functional scenarios.

The following points are covered in this paper: 1) propose a generic methodology to generate synthetic datasets for perception testing using logical scenario files, 2) implement a proof of concept of the proposed methodology using open source technologies and automotive standards and 3) test the proposed methodology to analyse the performance of 3D perception models under different scenarios and sensor setups. The rest of the paper is organised as follows: Section II provides background of the main blocks of the methodology, Section III includes a proof of concept implementation of the presented methodology using open-source technologies, Section IV takes advantage of the presented methodology from the point of view of a test engineer to test a 3D perception system under different sensor setups and Section V concludes the paper.

II. Data generation methodology

The main blocks of the proposed methodology can be seen in Fig. 1, which will be explained in the following subsections. The first input of this methodology consists of logical scenario description files, which are used as an input to generate concrete scenarios for its later simulation. These simulations can be replayed in a deterministic manner for open-loop simulations, which allows loading different sensor setups in the exact same virtual scenario but also allowing to modify certain conditions of the simulation (i.e., weather conditions). Finally, the methodology generates a labelled dataset with the raw sensor data that can be used to test perception systems.

A. Scenario Generation

Scenario generation consists in the process of generating concrete scenarios from logical scenario descriptions. These techniques have usually been classified into two types of methods [15]; data-driven techniques [16], which analyse traffic data or accident databases to infer relevant scenarios, and expert knowledge-driven techniques, where scenarios are carefully analysed and defined by experts. In [17], they perform a systematic literature review in scenario generation techniques and propose three new category types as not all the current methods can be classified correctly in the initial binary classification. The three new categories include random scenarios, in which parameters defining the scenario are randomly sampled; combinational scenario generation methods, in which scenarios are decomposed into atomic pieces that can be combined; and optimization-based scenario parameter selection and evaluation techniques. This methodology is not dependant on the selected scenario parameter method selected, however, the initial input of the methodology depends on the logical scenario description files. The ASAM OpenSCENARIO XML[3] format has been the main scenario description format used in the automotive industry. The newest versions of the standard allows defining scenarios at logical and concrete levels of abstraction. The newest ASAM OpenSCENARIO DSL[4] format defines a

[3]https://www.asam.net/standards/detail/openscenario-xml/
[4]https://www.asam.net/standards/detail/openscenario-dsl/

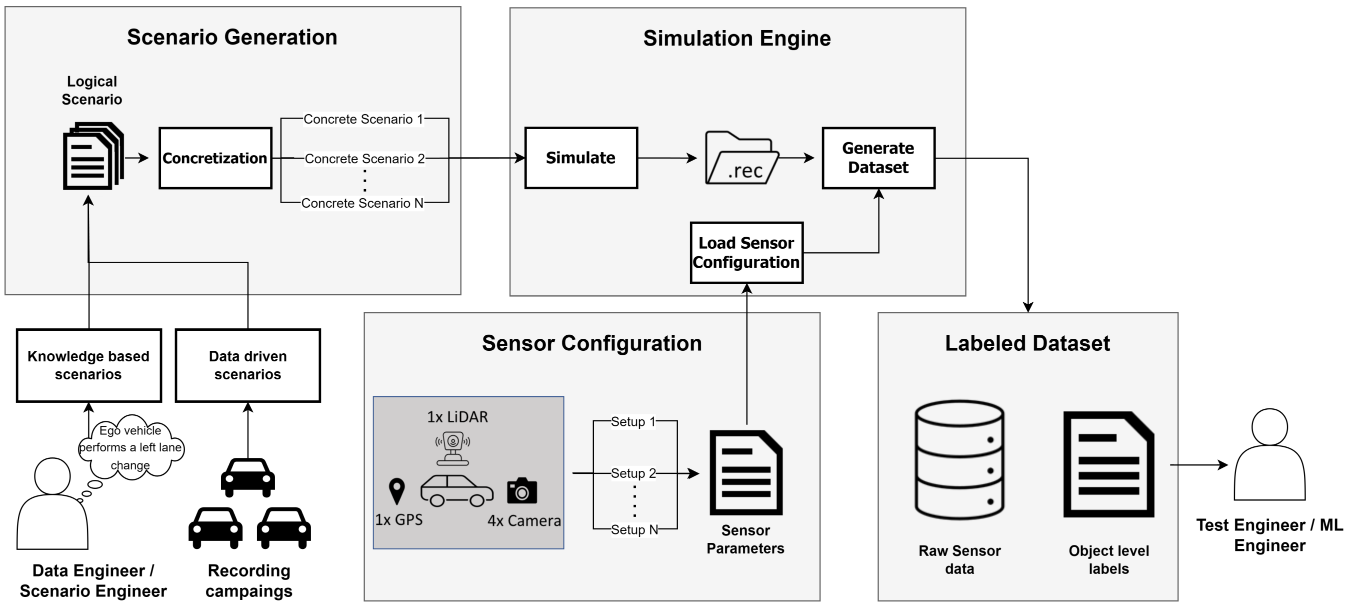


Fig. 1. Proposed synthetic dataset generation methodology using logical scenarios.

Manuscript 179 submitted to 2025 36th IEEE Intelligent Vehicles Symposium (IV). Received January 30, 2025.

domain-specific programming language to describe scenarios at all the previously presented levels of abstraction. Similarly, the three available versions of Scenic [18]–[20] define not only a logical scenario description language but also a compiler for the concretization step involved in scenario generation compatible with multiple automotive simulators. Lastly, [21] presents a two-level of abstraction Scenario Description Language (SDL) where the first level corresponds to functional scenarios and the second level is compatible with logical and concrete levels of abstraction. These scenario description files are usually tied to road topology description files to define the road network of a given scenario. Both OpenSCENARIO and Scenic are compatible with ASAM OpenDRIVE [5], a standardised definition file for defining road networks. Selecting a scenario description format to cover the static environment (road data) and dynamic environment (actor definition and behaviour) is crucial for the simulation step of the methodology, as not all automotive simulation engines are compatible or provide tools to execute certain scenario formats.

B. Simulation Engine

Simulation engines play a crucial role in the development and testing of ADS, providing a synthetic environment where various scenarios can be generated and evaluated. Among the existing simulation engines, Unreal Engine and Unity are prominent examples. These engines offer powerful tools for creating highly realistic environments with detailed graphics, physics simulations, and interactive elements. They have been widely used in the gaming industry and have also found applications in autonomous driving research by enabling the simulation of driving environments, including sensor simulation, vehicle dynamics, and physical 3D environments.

In the automotive open-source simulation domain, CARLA [22] and LGSVL [23] have emerged as prominent tools tailored for autonomous driving research, based on Unreal and Unity respectively. Leveraging the robust capabilities of their base simulation engines, these simulators offer a customizable platform to explore autonomous vehicle technologies, including support for various sensor models, vehicle dynamics, and environmental conditions. Equipped with a comprehensive suite of sensors such as cameras, LiDAR, and radars, both platforms facilitate thorough data collection and analysis, enabling researchers to improve their ADS iteratively. However, one of the possible drawbacks of these automotive simulators is that they are not directly compatible with standard scenario formats. CARLA provides the ScenarioRunner[6] tool which has a great level of support with the two OpenSCENARIO versions, however, it is not yet fully compatible with them, which can cause problems during the simulation of fully compliant OpenSCENARIO files. Other alternatives such as ESMINI[7], provide a way of graphically validating OpenSCENARIO XML files by visualizing the behaviour of the defined scenario, but is not meant for high-fidelity simulation. The following points need to be addressed before choosing a simulation engine:

- Select a simulation engine that covers the simulation needs of the perception system under test, including, the quality of the virtual environment and that the desired sensors can be simulated.
- Ensure that the simulation engine is compatible or can reproduce the selected scenario description formats.

C. Sensor Configuration

Regardless of the simulation engine, it is crucial to define the sensors' setup separately. This includes specifying their intrinsic parameters, which relate to their internal geometric configuration, in the case of cameras, these would include parameters of the camera sensors (e.g. focal length, distortion coefficients, etc.). It also involves determining their pose in terms of translation and rotation relative to a reference frame, such as the ego-vehicle rear axle projected to the ground plane (as specified in ISO-8855). Keeping a well-defined description of sensors and their characteristics enables the replication of different scenario setups and conditions using the same configurations which is a key component of any SBT pipeline used to evaluate perception systems. This also opens the possibility of performing optimal configuration searches on different scenarios.

The absence of a widely accepted standard for configuring sensor setups results in each database or pipeline devising its own way to define them. For example, in the Robotic Operating System (ROS), camera sensor poses are described using a transformation tree, and their intrinsic parameters are stored as a *CameraInfo* ROS message. Datasets such as KITTI simply lists both intrinsic and extrinsic parameters in a plain text file, NuScenes provides a relational database where they store the extrinsic and intrinsic values of the sensors for each recording and the Waymo dataset stores this data as protobuf files. To tackle this inconsistency, ASAM has introduced the ASAM OpenLABEL[8] standard, which provides a format and guidelines for labelling multi-sensor data, from the raw sensor data to object level and tag annotations, that facilitates the interfacing between real world and virtual testing environments.

D. Labelled Dataset

There are several open-source datasets of automotive labelled data available with several terabytes of sensor data, recorded both in the real world and in virtual environments on high-fidelity automotive simulation software. These datasets largely vary in terms of the sensor setups and scenarios recorded.

Unfortunately, there is no widely accepted standard data format for these kinds of datasets. Each of them utilises their own naming conventions, coordinate system frames, etc. Format inconsistencies like these naturally hinder the advance of the state of the art due to having to deal with different parsing

[5]https://www.asam.net/standards/detail/opendrive/
[6]https://github.com/carla-simulator/scenario_runner
[7]https://github.com/esmini/esmini
[8]https://www.asam.net/standards/detail/openlabel/

strategies, and so, there has been some effort in defining a standardised format for automotive datasets. For example, the OMEGA format which was developed in the context of the VVM project [9] and aims to be an standard format for storing reference and perception data. Another alternative is the previously introduced ASAM OpenLABEL standard, whose schema incorporates a large variety of annotation capabilities, such as multi-sensor object level annotations and scenario tagging.

III. IMPLEMENTATION

In this work, we have implemented a proof of concept of this methodology selecting a set of open-source technologies and standard annotation formats.

A. Scenario generation

For the scenario generation step, the second version of the Scenic [19] description language has been used. By defining logical scenario files using the syntax of the Scenic programming language, rules, parameter ranges and requirements can be set not only to define the desired road characteristics but also to define the parameter ranges for each actor in the scenario and their desired behaviour. One of the main advantages of Scenic is that it does not only provide the syntax for defining logical scenarios, but also deals with the parameter concretization step by sampling values and checking that they fulfil the desired parameter ranges and conditions. Scenic is also compatible with the ASAM OpenDRIVE[10] standard, widely used in the industry as a file description format to define road-networks with detail. Thus, the use of Scenic covers entirely the scenario generation block explained in Section II of the proposed methodology. Scenic is compatible with multiple simulation engines, including automotive simulators such as CARLA [22] or simulators oriented to robotics such as Webots [24].

B. Simulation engine

For our simulation engine, we have chosen CARLA [22], one of the main open-source automotive simulation engines based on Unreal Engine 4 that provides high-fidelity rendering. CARLA provides recording capabilities, which allows to register all the events and trajectories from all the actors and objects in a scenario in a recording file for its later replay. This feature enables one of the key points of the proposed methodology, allowing to decouple the target sensor setup from the simulation, being able to reproduce the same scenario with different sensor setups if needed. CARLA provides a Python API that can be used to access the state of all the objects of the simulation but also to spawn, attach, and retrieve sensors to the desired actor in the simulation. Among the sensors that can be simulated in CARLA we can find radars, LiDARs, IMUs, GNSSs and different types of cameras including RGB, Depth or Semantic segmentation cameras.

[9]https://www.vvm-projekt.de

[10]https://www.asam.net/standards/detail/opendrive/

C. Sensor configuration and dataset format

For setting up our sensors configurations, we have chosen the ASAM OpenLABEL format. This format is presented in a JSON file following a standardised structure with a JSON schema, making it easily interpretable by various tools and applications. Additionally, the OpenLABEL format facilitates scenario labelling by annotating contextual elements such as weather conditions, actions, or various objects present within the scene. To work on the OpenLABEL format we use the Video Content Description (VCD[11]) Python library [25]. This package enables an easy interaction with the OpenLABEL format and includes several useful utilities, such as performing coordinate frames transformations between sensors and annotated objects, and handle most of the workload from odometry changes. Concerning our use case, some common road-object classes are considered such as pedestrians, cars, trucks, buses, bicycles, and motorbikes. Moreover, apart from their class name and 3D position cuboid, both delta velocity and delta acceleration are annotated along with their respective vector magnitudes.

IV. EXPERIMENTS

In this section, we have generated a synthetic dataset following our proposed methodology. The dataset is composed of concrete scenarios generated from three different logical scenario definitions. The synthetic dataset has been used to test the performance of various state of the art LiDAR based 3D object detection models.

A. Selected Scenarios

We have selected three different logical scenarios from the provided NHTSA Pre-Crash scenarios [26] available in Scenic. An overview of the three logical scenario descriptions can be seen in Fig. 2.

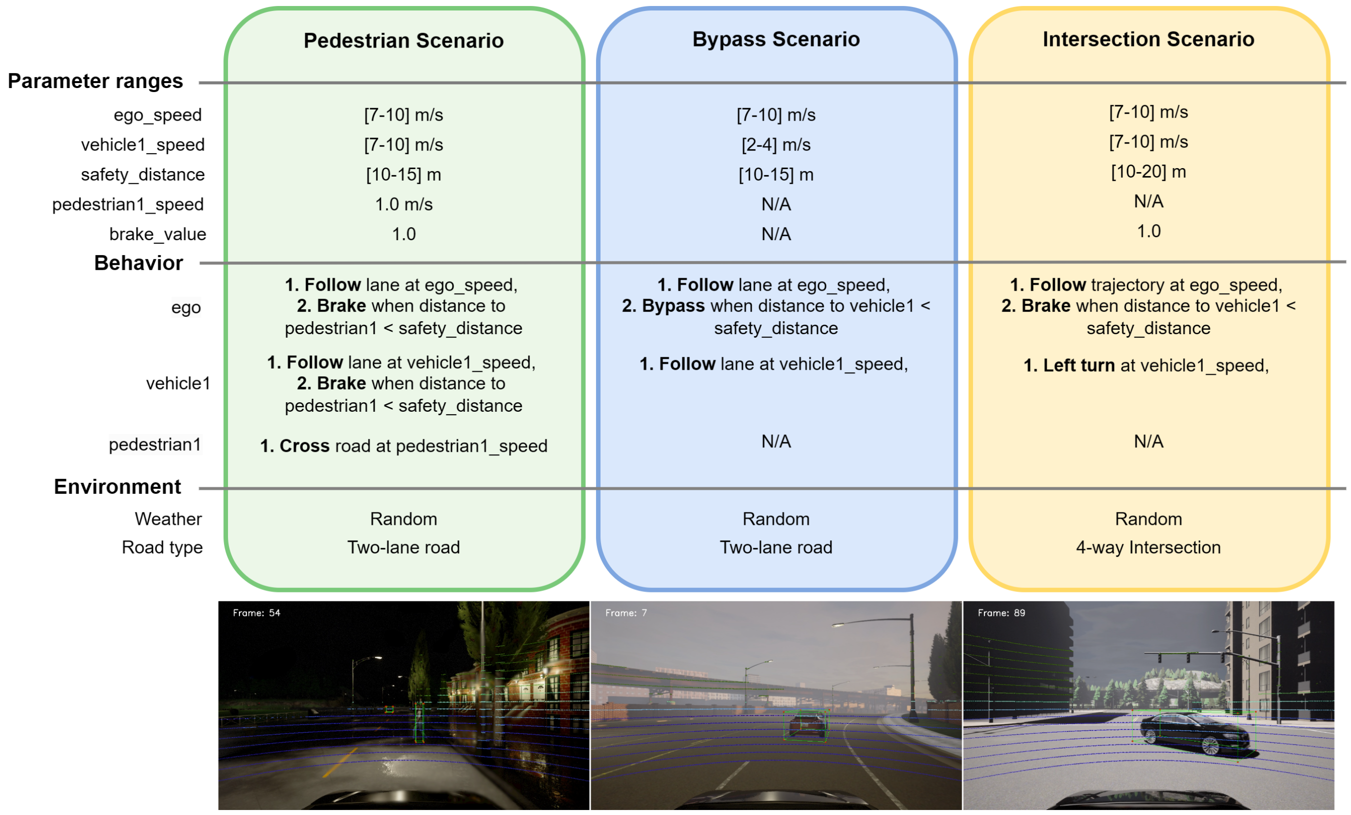
The first scenario involves three actors, the ego vehicle and another vehicle suddenly stopping due to a pedestrian unexpectedly crossing the street. In the bypassing scenario, the ego vehicle needs to perform an overtake by changing its lane to bypass a slower vehicle. Finally, in the intersection scenario, the ego vehicle must stop in a 4-way intersection when another vehicle performs a left turn in the intersection.

Each logical scenario has been concretised ten times, generating a sample dataset of 30 different concrete scenarios. In each concrete scenario not only the parameter ranges shown in Fig. 2 have been sampled, but also the 3D models of the actors involved in the simulations. Separating the sensor setup from the desired concrete simulation allows to simulate the three proposed sensor setups in a single simulation run instead of executing the same scenario each time a new sensor setup is needed.

B. 3D Perception system under test

With the resulting dataset, we have tested the performance of different 3D object detector models, more specifically PointPillars [27], Shape Signature Network (SSN) [28] and

[11]https://pypi.org/project/vcd/



	Pedestrian Scenario	Bypass Scenario	Intersection Scenario
Parameter ranges			
ego_speed	[7-10] m/s	[7-10] m/s	[7-10] m/s
vehicle1_speed	[7-10] m/s	[2-4] m/s	[7-10] m/s
safety_distance	[10-15] m	[10-15] m	[10-20] m
pedestrian1_speed	1.0 m/s	N/A	N/A
brake_value	1.0	N/A	1.0
Behavior			
ego	**1. Follow** lane at ego_speed, **2. Brake** when distance to pedestrian1 < safety_distance	**1. Follow** lane at ego_speed, **2. Bypass** when distance to vehicle1 < safety_distance	**1. Follow** trajectory at ego_speed, **2. Brake** when distance to vehicle1 < safety_distance
vehicle1	**1. Follow** lane at vehicle1_speed, **2. Brake** when distance to pedestrian1 < safety_distance	**1. Follow** lane at vehicle1_speed,	**1. Left turn** at vehicle1_speed,
pedestrian1	**1. Cross** road at pedestrian1_speed	N/A	N/A
Environment			
Weather	Random	Random	Random
Road type	Two-lane road	Two-lane road	4-way Intersection

Fig. 2. Overview of the parameter ranges, actor behaviors and environment conditions of the three selected NHTSA Pre-Crash scenarios.

CenterPoint [29]. All the models have been trained with labelled point cloud data from the NuScenes [2] dataset. The labelled point clouds have been captured using a Velodyne HDL-32 LiDAR which has 32 channels, a 360º horizontal FOV and a vertical FOV ranging from +10º to -30º. To evaluate these models against different sensor domains, we test their performances in the three scenarios defined in Fig 2 using the following sensor setups:

1) **Low-Density sensor:** 16 Channel LiDAR with a vertical Field Of View (FOV) of 15º to -15º.
2) **Ideal sensor:** 32 Channel LiDAR with a vertical FOV of 10º to -30º.
3) **High-Density sensor:** 64 Channel LiDAR with a vertical FOV of 10º to -30º.

In our evaluation, we apply the domain gap mitigation strategy proposed in [30] and calculate the true positive rate for each model against the dynamic objects defined in the scenarios. Based on the default evaluation metric of NuScenes [2], we consider a detection to be a true positive if it is located within a distance threshold of 0.5 meters of its ground truth, the obtained results can be seen in Table I.

For the car class, our analysis demonstrates that models achieve the best results when tested on the ideal sensor across all the defined logical scenarios. In this case, the PointPillars model obtains the best results for the bypassing and intersection scenarios, while SSN marginally outperforms the other two models in the pedestrian scenario.

In the case of PointPillars and SSN we can see how the models obtain better results using the low-density sensor when compared to the high-density sensor. This counter-intuitive effect is more accentuated in SSN and we could hypothesize that these models fail to generalize in denser point clouds. This is not the case for the CenterPoint model, where the high-density sensor obtains better results than its low-density counterpart proving that it generalizes better in denser point clouds but still achieving the best results using the ideal sensor configuration. For the pedestrian class, we can see that all models obtained the best results using the low-density sensor. In this case, PointPillars also obtains the best results and is the most consistent model across all sensor configurations. In the case of SSN and CenterPoint, we can see a severe model performance degradation with the higher-density model. Obtaining better results with the low-density

TABLE I

TRUE POSITIVE RATES OF POINTPILLARS (PP), SHAPE SIGNATURE NETWORK (SSN), AND CENTERPOINT (CP) ACROSS VARIOUS SENSOR SETUPS AND SCENARIOS FOR THE 'CAR' AND 'PEDESTRIAN' CLASSES

Class: Car

Sensor	Low-Density			Ideal			High-Density		
Model	PP	SSN	CP	PP	SSN	CP	PP	SSN	CP
bypassing	0.81	0.76	0.69	**0.90**	0.80	0.73	0.76	0.48	0.71
intersection	0.87	0.86	0.73	**0.93**	0.92	0.79	0.81	0.61	0.77
pedestrian	0.67	0.67	0.68	0.87	**0.88**	0.87	0.61	0.47	0.83

Class: Pedestrian

Sensor	Low-Density			Ideal			High-Density		
Model	PP	SSN	CP	PP	SSN	CP	PP	SSN	CP
pedestrian	**0.55**	0.48	0.27	0.51	0.35	0.18	0.46	0.15	0.11

sensor was not an expected behaviour and showcases how the methodology can aid in identifying problems when testing perception systems. This behaviour may be caused by the different FOV configuration of the low density sensor being benefitial in the pedestrian scenario or by a domain gap of the synthetic data, however, it should be further analysed to draw a clear conclusion.

V. Conclusions

Scenario-based testing is a current trend in the automotive industry and further work is necessary, not only in methodologies to generate synthetic data, but also in using and proposing standardised data formats. We have presented an end-to-end methodology to generate labelled datasets for perception testing using logical scenario file descriptions and automotive standards. Thanks to decoupling the scenario simulation from the sensor configuration, the exact same scenarios can be tested under different sensor setups and scenario conditions (i.e. weather conditions) in open-loop simulation. This can help to identify sensor domain adaptation problems or to test the expected effectiveness of perception systems against unknown scenarios. This statement has been proved by testing the performance of three state-of-the-art lidar-based object detectors against three different sensor setups in different automotive scenarios.

References

[1] E. T. . 562:2019, "Intelligent Transport Systems (ITS);Vehicular Communications;Basic Set of Applications; Analysis of the Collective Perception Service (CPS);," ETSI, Tech. Rep. CPM, 2019.

[2] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," in *CVPR*, 2020.

[3] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, V. Vasudevan, W. Han, J. Ngiam, H. Zhao, A. Timofeev, S. Ettinger, M. Krivokon, A. Gao, A. Joshi, Y. Zhang, J. Shlens, Z. Chen, and D. Anguelov, "Scalability in perception for autonomous driving: Waymo open dataset," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[4] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *International Journal of Robotics Research (IJRR)*, 2013.

[5] N. Kalra and S. M. Paddock, *Driving to Safety: How Many Miles of Driving Would It Take to Demonstrate Autonomous Vehicle Reliability?* Santa Monica, CA: RAND Corporation, 2016.

[6] "Road vehicles –Funtional Safety," International Organization for Standardization, Geneva, CH, Standard, 2022.

[7] "Road vehicles –Safety of the intended functionality," International Organization for Standardization, Geneva, CH, Standard, 2022.

[8] A. Gaidon, Q. Wang, Y. Cabon, and E. Vig, "Virtual worlds as proxy for multi-object tracking analysis," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2016, pp. 4340–4349.

[9] Y. Cabon, N. Murray, and M. Humenberger, "Virtual kitti 2," 2020.

[10] A. Kloukiniotis, A. Papandreou, C. Anagnostopoulos, and E. al., "CarlaScenes: A Synthetic Dataset for Odometry in Autonomous Driving," in *CVPR Workshops*, jun 2022, pp. 4520–4528.

[11] S. Yogamani, C. Hughes, J. Horgan, G. Sistu, P. Varley, D. O'Dea, M. Uricár, S. Milz, M. Simon, K. Amende *et al.*, "Woodscape: A multi-task, multi-camera fisheye dataset for autonomous driving," *arXiv preprint arXiv:1905.01489*, 2019.

[12] E. De Gelder, J. P. Paardekooper, A. Khabbaz Saberi, H. Elrofai, O. op den Camp, S. Kraines, J. Ploeg, and B. De Schutter, "Towards an Ontology for Scenario Definition for the Assessment of Automated Vehicles: An Object-Oriented Framework," *IEEE Transactions on Intelligent Vehicles*, pp. 1–16, 2022.

[13] T. Menzel, G. Bagschik, Maurer, and Markus, "Scenarios for Development, Test and Validation of Automated Vehicles," *IEEE Intelligent Vehicles Symposium, Proceedings*, vol. 2018-June, pp. 1821–1827, 2018.

[14] C. Neurohr, L. Westhofen, M. Butz, M. H. Bollmann, U. Eberle, and R. Galbas, "Criticality Analysis for the Verification and Validation of Automated Vehicles," *IEEE Access*, vol. 9, no. i, pp. 18 016–18 041, 2021.

[15] D. Nalic, T. Mihalj, M. Bäumler, M. Lehmann, A. Eichberger, and S. Bernsteiner, "Scenario based testing of automated driving systems: A literature survey," Nov. 2020, p. 1, fISITA Web Congress 2020 ; Conference date: 24-11-2020 Through 24-11-2020. [Online]. Available: https://go.fisita.com/fisita2020

[16] J. Cai, W. Deng, H. Guang, Y. Wang, J. Li, and J. Ding, "A Survey on Data-Driven Scenario Generation for Automated Vehicle Testing," *Machines*, vol. 10, no. 11, 2022.

[17] L. Birkemeyer, C. King, and I. Schaefer, "Is scenario generation ready for sotif? a systematic literature review," in *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*, 2023, pp. 472–479.

[18] D. J. Fremont, X. Yue, T. Dreossi, A. L. Sangiovanni-Vincentelli, S. Ghosh, and S. A. Seshia, *Scenic: A language for scenario specification and scene generation*, 2019.

[19] D. J. Fremont, E. Kim, T. Dreossi, S. Ghosh, X. Yue, A. L. Sangiovanni-Vincentelli, and S. A. Seshia, "Scenic: a language for scenario specification and data generation," *Machine Learning*, vol. 112, no. 10, pp. 3805–3849, 2023. [Online]. Available: https://doi.org/10.1007/s10994-021-06120-5

[20] E. Vin, S. Kashiwa, M. Rhea, D. J. Fremont, E. Kim, T. Dreossi, S. Ghosh, X. Yue, A. L. Sangiovanni-Vincentelli, and S. A. Seshia, "3d environment modeling for falsification and beyond with scenic 3.0," in *Computer Aided Verification*, C. Enea and A. Lal, Eds. Cham: Springer Nature Switzerland, 2023, pp. 253–265.

[21] X. Zhang, S. Khastgir, and P. Jennings, "Scenario description language for automated driving systems: A two level abstraction approach," in *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2020, pp. 973–980.

[22] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "Carla: An open urban driving simulator," 2017. [Online]. Available: https://arxiv.org/abs/1711.03938

[23] G. Rong, B. H. Shin, H. Tabatabaee, Q. Lu, S. Lemke, M. Mozeiko, E. Boise, G. Uhm, M. Gerow, S. Mehta, E. Agafonov, T. H. Kim, E. Sterner, K. Ushiroda, M. Reyes, D. Zelenkovsky, and S. Kim, "LGSVL simulator: A high fidelity simulator for autonomous driving," *CoRR*, vol. abs/2005.03778, 2020. [Online]. Available: https://arxiv.org/abs/2005.03778

[24] O. Michel, "Webots: Professional mobile robot simulation," *Journal of Advanced Robotics Systems*, vol. 1, no. 1, pp. 39–42, 2004. [Online]. Available: http://www.ars-journal.com/International-Journal-of-Advanced-Robotic-Systems/Volume-1/39-42.pdf

[25] M. Nieto, O. Senderos, and O. Otaegui, "Boosting ai applications: Labeling format for complex datasets," *SoftwareX*, vol. 13, p. 100653, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352711020303666

[26] E. D. Swanson, F. Foderaro, M. Yanagisawa, W. G. Najm, and P. Azeredo, "Statistics of Light-Vehicle Pre-Crash Scenarios Based on 2011–2015 National Crash Data," *US DOT, National Highway Traffic Safety Administration*, no. August, p. 136, 2019.

[27] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 697–12 705.

[28] X. Zhu, Y. Ma, T. Wang, Y. Xu, J. Shi, and D. Lin, "Ssn: Shape signature networks for multi-class object detection from point clouds," in *Proceedings of the European Conference on Computer Vision*, 2020.

[29] T. Yin, X. Zhou, and P. Krähenbühl, "Center-based 3d object detection and tracking," *CVPR*, 2021.

[30] A. Iglesias, M. García, N. Aranjuelo, I. Arganda-Carreras, and M. Nieto, "Analysis of point cloud domain gap effects for 3d object detection evaluation," in *Proceedings of the 19th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 4: VISAPP*, INSTICC. SciTePress, 2024, pp. 278–285.

Manuscript 179 submitted to 2025 36th IEEE Intelligent Vehicles Symposium (IV).
Received January 30, 2025.