Description of Corner Cases in Automated Driving: Goals and Challenges

Daniel Bogdoll[1*]   Jasmin Breitenstein[2*]   Florian Heidecker[3*]   Maarten Bieshaar[3]
Bernhard Sick[3]   Tim Fingscheidt[2]   J. Marius Zöllner[1]

Abstract

Scaling the distribution of automated vehicles requires handling various unexpected and possibly dangerous situations, termed corner cases (CC). Since many modules of automated driving systems are based on machine learning (ML), CC are an essential part of the data for their development. However, there is only a limited amount of CC data in large-scale data collections, which makes them challenging in the context of ML. With a better understanding of CC, offline applications, e.g., dataset analysis, and online methods, e.g., improved performance of automated driving systems, can be improved. While there are knowledge-based descriptions and taxonomies for CC, there is little research on machine-interpretable descriptions. In this extended abstract, we will give a brief overview of the challenges and goals of such a description.

1. Introduction

Corner cases (CC) are data that occur infrequently or represent a critical situation and are only available in datasets to a limited extent, if at all. However, for machine learning (ML), CC are important as they are required for training, verification, and improved performance of ML models during inference within automated driving systems.

While automated vehicles will always encounter new CC, it is crucial to have training data with a wide variety of scenarios [18], including CC. This way, failures and unexpected behavior of ML models during inference can be reduced. Such failures can occur on several levels of the
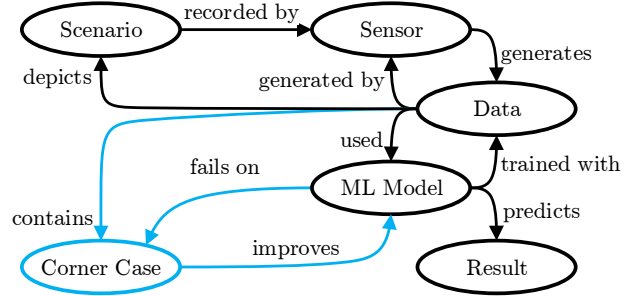

Figure 1: ML pipeline in automated driving with an emphasis on corner cases and their relation to data and ML models based on corner case descriptions.

system, such as missing detections or incorrect behavior predictions [16]. To achieve a good *online* performance of automated driving systems, it is necessary also to consider the previous *offline* development steps, such as dataset engineering. Since CC play a crucial role in both fields, corner case descriptions (CCD) act as a basis.

CC can be described either by experts or directly by the limits of a method [16]. On one hand, they can be divided into the *sensor layer*, *content layer* and *temporal layer*. *Sensor-layer* CC are directly connected to a sensor, e.g., dead pixels. The *content layer* ranges from domain shifts and object-related CC to events on the scene[1] level. Finally, the *temporal layer* includes all scenario-related CC.

On the other hand, *method-layer* CC occur when an ML model reaches its limits, as when failures occur due to insufficient model knowledge caused by incomplete training data [16]. Even models as DeepXplore [26], which automatically create CC, are unable to create descriptions of those.

Describing CC with a proper description language and using those descriptions are fields with many challenges and applications. CCD are necessary for different aspects of the *offline* development pipeline in automated driving and the subsequent *online* validation and deployment tasks.

A major research question is how to provide a valid description of CC in co-existence with established scenario

[1]Daniel Bogdoll and J. Marius Zöllner are with Technical Cognitive Systems, FZI Research Center for Information Technology, Schönfeldstraße 8, 76131 Karlsruhe, Germany, {bogdoll, zoellner}@fzi.de

[2]Jasmin Breitenstein and Tim Fingscheidt are with Institute for Communications Technology, Technische Universität Braunschweig, Schleinitzstraße 22, 38106 Braunschweig, Germany, {j.breitenstein, t.fingscheidt}@tu-bs.de

[3]Florian Heidecker, Maarten Bieshaar and Bernhard Sick are with Intelligent Embedded Systems, University of Kassel, Wilhelmshöher Allee 73, 34121 Kassel, Germany, {florian.heidecker, mbieshaar, bsick}@uni-kassel.de

*These authors contributed equally

[1]We follow the definitions of scene and scenario by Ulbrich et al. [31], where a scene is a snapshot, and a scenario consists of successive scenes

description languages. This can be useful in *offline* applications to develop automated driving systems: First, existing data can be automatically described based on a CCD, including the coverage of datasets in respect to CC. Valid descriptions of CC are necessary to generate synthetic data or to record real data. Subsequently, CCD can be utilized for *online* applications. In the validation of automated driving systems, CCD can simulate variations of certain CC to cover as many variants of a situation as possible. During the deployment of automated vehicles, perception and fusion can benefit from such descriptions to determine which sensors are best equipped to detect certain CC, or even serve as an interface to the prediction or planning tasks.

An insight into how we envision CC within a description language is shown in Figure 1. Scenarios, either conceptual or from the real world, are embedded within the data recorded by sensors. Only here can CC be detected. An ML model to perform perception tasks might fail online on CC data, but can be improved, if CC can be detected and properly understood. Therefore, CCD have the potential to increase the performance of ML models.

Such a CCD comprises various aspects, some of which are already contained in established scenario description languages [4, 25]. However, there are significant research gaps regarding CCD which we discuss in the following.

Our contributions of this extended abstract are: We provide an overview over the state of the art of multiple research areas related to CC and CCD. We formulate six research questions to outline topics for future work, and describe the potential of CCD based on three use cases in regard to our research questions.

The remainder of this article is structured as follows: Section 2 provides an overview of related research. Section 3 formulates our research questions. Section 4 includes use cases. Finally, Section 5 provides a summary and an outlook.

2. Related Work

So far, CCD have received little attention in research. However, they have many connections to other research areas, such as scenario descriptions, which are used in validation processes to generate suitable validation scenarios, and are currently limited in their ability to describe CC. In addition, there are related research areas such as CC detection, generation, and dataset engineering that benefit from established CCD.

2.1. Scenario Description

In the field of automated driving systems, all real world data is embedded into scenarios, even though CC can also occur and be detected on lower levels [16]. Scenarios can be described on different levels of abstraction, namely *functional, logical and concrete* [23]. In the following, we will give an overview of scenario descriptions as we aim to combine existing scenario descriptions with a CCD. Since every CC stems from a scenario, as shown in Figure 1, scenario description is the field with the closest relation to CCD. Here, the basic entities in a scene are described by concrete scenario description languages. A CCD expands on this and describes the CC, based on the categories of [16]. One example might be to classify a scene or scenario by a CCD with a level, a type, and a machine-interpretable description, including relevant entities.

2.1.1 PEGASUS Layer Model

The PEGASUS model [25, 32], a generic description model for automated driving scenarios, contains six layers, ranging from road-level to digital information. The first layer deals with road geometry and its characteristics, while the second layer includes traffic infrastructure. Temporary modifications are considered in the third layer, such as movable road infrastructure. All other movable objects, vehicles, and vulnerable road users (VRU) are described in the fourth layer. Environmental influences, e.g., weather, are summarized in the fifth layer. Finally, the sixth layer deals with digital information and describes V2X (Vehicle-to-everything) information and digital maps. The language aims to describe scenarios on all three abstraction levels [23].

2.1.2 Open Scenario & Open Drive

The key feature of OpenScenario is to describe traffic scenarios consisting of multiple actors, focusing on concrete scenarios. Thereby, the temporal succession of actions such as an overtaking maneuver is clearly described. A standard for the scenario description is presented in ASAM OpenScenario [4]. However, OpenScenario does not provide a driver model or advanced motion dynamics for vehicles. ASAM OpenDrive [3] is a standard and describes the road network, which contains the road geometry, lanes, and signals. The surface profile of the road is standardized within ASAM OpenCRG (Curved Regular Grid) [2]. Combining these components, automated in-the-loop vehicle simulation of dynamic and static content is possible.

2.1.3 Ontology-Based Descriptions

Ontologies have the ability to inherit structure and terminology from taxonomies to describe relations, attributes, and general concepts of and between classes. As shown in [14], their representations can vary from meta-levels to very detailed, even perception-related attributes and therefore seem suited for all levels. As demonstrated in [22], they can be used to describe traffic scenes, including temporal context.

This way, dynamic objects and details about their configurations, intentions, and information about the road layout can be combined for a holistic scenario description. Once an ontology is designed, instances can be created to describe concrete situations.

2.1.4 Operational Design Domain

An operational design domain (ODD) describes the operating domain of an automated driving system, that has been designed to limit the situations that possibly occur. This can act as a baseline to take into account combinations of possible operating conditions during validation [21]. Morphological boxes, also called *zwicky boxes*, provide a possibility to describe an ODD using possible dimensions, i.e. relevant features, and corresponding attributes, i.e. value ranges [5, 15]. Describing CC in an ODD is difficult as many per definition would not be included in the description in the first place. In the case of *zwicky boxes*, combinatorial anomalies can be created, but, e.g., any CC based on temporal context is out of range for the description.

2.2. Corner Case Detection and Generation

The safety of automated driving functions depends highly on reliable CC detection both for curating suitable training and test datasets and as an online alert system.

Corner Case Detection: Depending on conceptual levels [16, 8], different CC exists and, hence, implicating different detection methods. Certain types of CC can be detected from single image frames, such as physical-level CC, e.g., glare situations [17, 12], or content-layer CC, e.g., unknown objects in the traffic situation [33]. Temporal-layer CC encompass an entire time span. Hence, detection methods need to take temporal data into account, such as video sequences [7]. Such detection systems typically provide metric-based assessments of situations, such as shown in [8]. Such metrics, without a context, typically do not contribute to an improved understanding of the situation. A CCD is a potent tool to embed such metrics in a context.

Corner Case Generation: CC generation strategies can be divided into the classes *model-based, data-driven and scenario-based* [29]. Tuncali et al. [29] propose a model-based generation method that is able to generate CC based on a combination of covering arrays and requirement falsification methods. Chou et al. [10] propose a similar system, which determines corner case situations for an automated vehicle which are critical, but not impossible to handle. Data-driven methods, such as shown in [24], rely on existing datasets and samples from the available data. Möller et al. [24] generate potential CC by learning a latent space based on available data and subsequently gen-

erate out-of-domain scenarios, which are still close to the actual in-distribution samples and therefore likely to be realistic. Finally, scenario-based methods mostly utilize descriptions of experts or compiled within accident reports. Pretschner et al. [27] propose a *simulation-based heuristic search* to generate similar scenarios as shown in [10], which are basically critical, but do not necessarily lead to a collision. [19] shows a methodology, which samples CC based on an ontology, which describes the environment of a vehicle. CC generation is dependent on sufficient and precise CCD to generate suitable data.

2.3. Dataset Engineering

As dataset engineering, we identify the task of putting together a suitable dataset for the underlying perception function. When recording new data, CCD can either act as a trigger a recording during fleet operation [18] or as a formal description for controlled records. When considering existing datasets, the challenge is to describe them, quantify their coverage, and measure their diversity. Sadat et al. [28] define various complexity measures for dataset diversity, taking into account the relevance of certain traffic situations. In *active learning*, one selects interesting data for additional labeling to keep costs of labeling low. As an example of using CC in this research area, Dhananjaya et al. [12] applied active learning for weather-related CC. In contrast to general dataset coverage, such active learning methods are based on the underlying model [28]. CCD can aid the task of dataset engineering as they give a way to describe the CC coverage of a dataset and to filter for certain CC. This then allows determining the CC coverage of ML models based on their training dataset.

3. Research Questions

The concept of CCD has to overcome various challenges before it can leverage its full potential. We summarize these points with the following research questions (RQ), focusing on the description itself, followed by *offline* and subsequent *online* applications.

Description (*RQ1*): *How to describe CC on different levels of abstraction?* – Heidecker et al. [16] present a high-level categorization of CC in different layers and levels. We aim to combine this with a concrete scenario description language, as outlined in Section 2.1, or meta-data to build a complete description of CC. This makes it possible for us to use existing elements of scenario description languages, which describe the scene in its basic elements. We believe an add-on should expand on this and address the CC, whereby we differentiate between the categories of [16] in the description, and then specifying the details. To improve and complete the description languages, data-driven approaches should be utilized, e.g. adding missing object classes. Based on a CCD, we see

multiple *offline* applications, which we address with the following RQ:

Automation (*RQ2*): *How to automatically describe data that contain CC?* – For an efficient generation of scenarios containing CC, and other tasks, it is important to not only have a valid CCD but to be able to extract them automatically. Methodologies to detect CC on sensor and object data levels are shown in [8, 30]. Having a machine-readable CCD as an interface will allow transferring algorithmic outputs into a CCD automatically. Thus, we can analyze existing datasets and enhance them with CCD for every single frame. While there is early research in this field, no extensive methodology has been developed to describe CC precisely. Combining existing CC levels [16] with efficient scenario description languages such as *zwicky boxes*, we aim to provide an automatic way to describe CC.

Data Generation (*RQ3*): *How to generate or record CC from descriptions?* – CCD will be used to build up scenario catalogs as concrete guidelines for human-performed data recordings or as input for data augmentation tools, such as [20], or simulation environments, e.g., CARLA [13]. They are also potent trigger descriptions for fleet recordings [18]. Feeding automatically generated CCD from RQ2 into a simulation tool, enables automatic description and generation of corner cases.

Dataset Analysis (*RQ4*): *How to determine the coverage of CC in training data?* – We aim to describe the coverage of a CC detector or ML model. If we describe the training/testing data with a CCD (e.g., by including the context), this will represent the content and thus the cases covered in the dataset. Transferring this description to the ML model trained and tested on the dataset, we can obtain the coverage that the ML model has or should have via evaluating its performance achieved on that specific dataset and coverage. Conversely, we also know what it does not cover. Additional training data would now be selected to achieve better generalization and coverage of the ML model. In case some CC are missing to fulfill the required coverage, the CCD is used to explicitly record or generate these CC scenarios synthetically, see *RQ3*. These *offline* development stages enable the *online* deployment of automated vehicles in the real world, which we address with the following research questions:

Automated Driving (*RQ5*): *How to utilize CCD for driving automation?* – CCD can be utilized to provide proper situation understanding for perception tasks, such as sensor fusion, since not all sensors are equally well suited to detect CC. CCD as a machine-readable format is an ideal candidate as an interface between the perception and later stages of the automated driving pipeline, transferring CC-related information to models such as prediction or plan-

ning. One example is to describe CC based on relevant objects instead of a static metric, which is a more actionable interface. While not much research regarding this topic can be found in academia, in industry, the activation of remote assistance is often triggered as soon as CC are detected or the ODD is exited [1].

Testing (*RQ6*): *How to validate and verify ML systems based on CCD?* – Scenario description languages are currently also used for automated testing and verification while not providing sufficient information about the coverage of CC within these scenarios. With CCD as an extension, this will become possible, improving upon existing approaches.

4. Addressed Use Cases

The CCD and associated RQ show why a description language and the corresponding structuring of CC are necessary. To demonstrate the potential of a CCD, we build upon the RQ and show their application-related relevance in the real world.

Figure 2: A visually challenging scene due to glare [16] (l) and an unknown object in the field of view [9] (r).

For this reason, we present the following three use cases and show how they would benefit from the CCD. We sketch out conceptual CCD for the use cases, focused on VRU.

4.1. Glare Effect

Being blinded by the sun, an artificial light source, or even reflected light, hence not seeing the actual object (e.g., pedestrian), is a situation that everyone is familiar with (see Figure 2 (l)). ML models that work with image data also have problems with being blind, as it harms the object detection or prediction of any kind. The glare can lead to a failure of the driving function and, in the worst case, to an accident. First, a CCD is necessary to allow tailored handling of the situation. A CCD including a scenario description, as in *RQ1*, of Figure 2 (l) would conceptually be: CC-level - physical level, CC-type - global outlier, scene - winter, sun - low (over the road), visibility - poor, vehicles -

Figure 3: Sequence of temporarily occluded road users [11].

yes, vehicle1: - on the other lane, coming in our direction, VRU - yes, VRU1: type - pedestrian, located - ahead on the road, moving - from left to right. The combinatorics of an extensive CCD allow generating new scenarios with glare effects *RQ3*. Regarding *RQ5*, based on the CCD, the sensor fusion layer could put more emphasis on other sensors, such as LiDAR or RADAR, as the system has gained an awareness that the image data has currently reduced information richness. A CCD can also be used to describe test cases that an ML model must pass for the validation *RQ6*.

4.2. Unknown Objects

In road traffic, an unmanageable number of scenarios occur every day. Some of these scenarios contain objects (see Figure 2 (r)), which were never or only to a small extent included in the training data of an ML method. In addition, new objects are constantly being created and modified, which negatively affects object detection reliability and leads to an increased risk of accidents when used in autonomous driving functions. According to [16], this group can be described as object-level CC and is included in the content layer. The challenge is to identify these CC and make them manageable for the driving function. A precise CCD *RQ1* can have a positive influence on object detection in several ways. A conceptual description of the CC in Figure 2 (r) is: CC-level - object level, CC-type - single-point anomaly, VRU - yes, VRU1: type - pedestrian, located - next to the road, moving - no, clothing - costumed, direction of view - towards us.

Blum et al. [6] demonstrate, how *RQ3* can be addressed with automated CC generation for unknown objects, resulting in similar data as shown in Figure 2 (r). With *RQ4*, based on a CCD, the typically broad category of unknown objects can be improved drastically, leading to a much better understanding of such objects. This would be convenient for object detection since we can quantify the ML model's coverage via the description. Also, CC are important for unknown objects because they are needed to validate the ML Model (*RQ6*).

4.3. Occluded Objects

Situations concerning temporarily occluded VRU are often accompanied by an increased risk for collision due to the uncertainty about the VRU reappearance. As such a situation covers an entire period, it describes a CC on the temporal layer [16]. A conceptual description of the presented scenario in Figure 3 would be: CC-level - scenario level, CC-type - risky scenario, infrastructure - intersection, vehicles - yes, vehicle1: - coming from left and left turn, VRU - yes, VRU1: type - pedestrian, located - opposite street side, moving - crossing from right to left, visible - true. This CCD applies to the left and right images in Figure 3. For the middle image, the scene description would not contain a pedestrian even though we know one is present because temporal context is missing. The CCD now changes to visible - false but keeps the pedestrian as an entity.

The challenge is to ensure no critical situations appear, which could not be handled by the automated driving system. Based on an automatically created CCD (*RQ1*, *RQ2*), the driving function (*RQ5*) would be aware of the occluded VRU. A CCD also allows the automatic generation of synthetic scenarios with occluded VRUs (*RQ3*).

5. Conclusion and Outlook

In this work, we formulated research questions for the application and need for valid corner case descriptions (CCD) focusing on machine learning (ML) models. In this regard, we reviewed existing scenario description languages based on their ability to describe corner cases (CC) and related research fields, such as CC generation. Based on our formulated research questions, we have shown the potential of a CCD based on three challenging situations within the domain of automated driving in the real world. We propose a combination of high-level knowledge-based CCD, low-level scenario-based CCD, and metric-based CC severity assessments as the basis for a machine-interpretable CCD, which will be the focus of future research.

Acknowledgment

References

[1] E. Ackerman. What Full Autonomy Means for the Waymo Driver. *IEEE Spectrum*, 2021. 4

[2] ASAM e. V. ASAM OpenCRG, 2021. 2

[3] ASAM e. V. ASAM OpenDRIVE, 2021. 2

[4] ASAM e. V. ASAM OpenSCENARIO, 2021. 2

[5] M. Blitzer, M. Herrmann, and E. Mayer-John. System Co-Design (SCODE): Methodology for the Analysis of Hybrid Systems. *at - Automatisierungstechnik*, 68(6):488–499, 2020. 3

[6] H. Blum, P. Sarlin, J. Nieto, R. Siegwart, and C. Cadena. Fishyscapes: A Benchmark for Safe Semantic Segmentation in Autonomous Driving. In *Proc. of ICCV - Workshops*, pages 2403–2412, Seoul, Korea, 2019. 5

[7] J.-A. Bolte, A. Bär, D. Lipinski, and T. Fingscheidt. Towards Corner Case Detection for Autonomous Driving. In *Proc. of IV*, pages 438–445, Paris, France, 2019. 3

[8] J. Breitenstein, J.-A. Termöhlen, D. Lipinski, and T. Fingscheidt. Systematization of Corner Cases for Visual Perception in Automated Driving. In *Proc. of IV*, pages 1257–1264, Las Vegas, NV, USA, 2020. 3, 4

[9] R. Chan, K. Lis, S. Uhlemeyer, H. Blum, S. Honari, R. Siegwart, M. Salzmann, P. Fua, and M. Rottmann. Segmentmeifyoucan: A benchmark for anomaly segmentation. *arXiv preprint:2104.14812*, pages 1–20, 2021. 4

[10] G. Chou, Y. Sahin, L. Yang, K. Rutledge, P. Nilsson, and N. Ozay. Using control synthesis to generate corner cases: A case study on autonomous driving. *IEEE Trans. on CAD*, 37(11):2906–2917, 2018. 3

[11] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The Cityscapes Dataset for Semantic Urban Scene Understanding. In *Proc. of CVPR*, pages 3213–3223, Las Vegas, NV, USA, 2016. 5

[12] M. Dhananjaya, V. Kumar, and S. Yogamani. Weather and Light Level Classification for Autonomous Driving: Dataset, Baseline and Active Learning. *arXiv preprint:2104.14042*, pages 1–6, 2021. 3

[13] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun. CARLA: An Open Urban Driving Simulator. *arXiv preprint:1711.03938*, pages 1–16, 2017. 4

[14] B. El Asmar, S. Chelly, N. Azzi, L. Nassif, J. Asmar, and M. Färber. AWARE: A Situational Awareness Framework for Facilitating Adaptive Behavior of Autonomous Vehicles in Manufacturing. In *Proc. of ISWC*, pages 651–666. Athens, Greece, 2020. 2

[15] C. Gladisch, C. Heinzemann, M. Herrmann, and M. Woehrle. Leveraging Combinatorial Testing for Safety-critical Computer Vision Datasets. In *Proc. of CVPR Workshops*, pages 1–8, Seattle, WA, USA, 2020. 3

[16] F. Heidecker, J. Breitenstein, K. Rösch, J. Löhdefink, M. Bieshaar, C. Stiller, T. Fingscheidt, and B. Sick. An Application-Driven Conceptualization of Corner Cases for Perception in Highly Automated Driving. pages 1–8, 2021. 1, 2, 3, 4, 5

[17] I. Jatzkowski, D. Wilke, and M. Maurer. A Deep-Learning Approach for the Detection of Overexposure in Automotive Camera Images. In *Proc. of ITSC*, pages 2030–2035, Maui, HI, USA, 2018. 3

[18] A. Karpathy. CVPR 2021 Workshop on Autonomous Driving Tesla Keynote, 2021. Accessed 9-July-2021. 1, 3, 4

[19] F. Klueck, Y. Li, M. Nica, J. Tao, and F. Wotawa. Using Ontologies for Test Suites Generation for Automated and Autonomous Driving Functions. In *Proc. of ISSRE Workshops*, pages 118–123, Memphis, TN, USA, 2018. 3

[20] T. Koduri, D. Bogdoll, S. Paudel, and G. Sholingar. AUREATE: An Augmented Reality Test Environment for Realistic Simulations. In *Proc. of WCX*, pages 1–9, Detroit, MI, USA, 2018. 4

[21] P. Koopman and F. Fratrik. How Many Operational Design Domains, Objects, and Events? In *Proc. of AAAI Workshops*, pages 1–4, Honolulu, HI, USA, 2019. 3

[22] F. Kuhnt. *Holistic Temporal Situation Interpretation for Traffic Participant Prediction*. PhD thesis, Karlsruhe Institute of Technology, 2020. 2

[23] T. Menzel, G. Bagschik, and M. Maurer. Scenarios for Development, Test and Validation of Automated Vehicles. *arXiv preprint:1801.08598*, pages 1–7, 2018. 2

[24] F. Möller, D. Botache, D. Huseljic, F. Heidecker, M. Bieshaar, and B. Sick. Out-of-distribution Detection and Generation Using Soft Brownian Offset Sampling and Autoencoders. pages 1–10, 2021. 3

[25] PEGASUS Project. PEGASUS Method: An Overview, 2019. 2

[26] K. Pei, Y. Cao, J. Yang, and S. Jana. DeepXplore: Automated Whitebox Testing of Deep Learning Systems. In *Proc. of SOSP*, pages 1–18, Shanghai, China, 2017. 1

[27] A. Pretschner, F. Hauer, and T. Schmidt. Tests für Automatisierte und Autonome Fahrsysteme. *Informatik Spektrum*, 44(3):214–218, 2021. 3

[28] A. Sadat, S. Segal, S. Casas, J. Tu, B. Yang, R. Urtasun, and E. Yumer. Diverse Complexity Measures for Dataset Curation in Self-Driving. *arXiv preprint:2101.06554*, pages 1–13, 2021. 3

[29] C. Tuncali, G. Fainekos, D. Prokhorov, H. Ito, and J. Kapinski. Requirements-Driven Test Generation for Autonomous Vehicles With Machine Learning Components. *IEEE Trans. on Intelligent Vehicles*, 5(2):265–280, 2020. 3

[30] L. Töttel, M. Zipfl, D. Bogdoll, M. Zofka, and J. Zöllner. Reliving the Dataset: Combining the Visualization of Road Users' Interactions with Scenario Reconstruction in Virtual Reality. *arXiv preprint:2105.01610*, pages 1–8, 2021. 4

[31] S. Ulbrich, T. Menzel, A. Reschka, F. Schuldt, and M. Maurer. Defining and Substantiating the Terms Scene, Situation, and Scenario for Automated Driving. In *Proc. of ITSC*, pages 982–988, Las Palmas de Gran Canaria, Spain, 2015. 1

[32] H. Weber, J. Bock, J. Klimke, C. Roesener, J. Hiller, R. Krajewski, A. Zlocki, and L. Eckstein. A Framework for Definition of Logical Scenarios for Safety Assurance of Automated Driving. *Traffic Injury Prevention*, 20:65–70, 2019. 2

[33] Y. Xia, Y. Zhang, F. Liu, W. Shen, and A. Yuille. Synthesize then Compare: Detecting Failures and Anomalies for Semantic Segmentation. In *Proc. of ECCV*, pages 1–17, Glasgow, UK, 2020. 3