Simulation Driven Design and Test for Safety of AI Based Autonomous Vehicles

Vasu Singh, Siva Kumar Sastry Hari, Timothy Tsai, Mandar Pitale
NVIDIA

Abstract

An autonomous vehicle (AV) integrates sophisticated perception and localization components to create a model of the world around it, which is then used to navigate the vehicle safely. Machine learning (ML) based models are pervasively used in these components to extract object information from noisy sensor data. The requirements for these components are primarily set to achieve as high accuracy as possible. With modern AVs deploying many sensors (cameras, radars, and LiDARs), processing all the data in real-time leads to engineers making trade-offs which might result in a sub-optimal system in certain driving situations. Due to the lack of precise requirements on individual components, modular testing and validation also becomes challenging.

In this paper, we formulate the problem of deriving abstract world model accuracy needed for safe AV behavior from top level driving scenario simulations. This is computationally expensive as the world model can contain many objects with several attributes and an AV extracts a world model every time-step during the simulation. We describe approaches to efficiently address the problem and derive component-level requirements and tests.

1. Introduction

The field of autonomous driving is rapidly evolving with the advancement of sensor and computing technologies. Establishing safety of AVs is a challenging endeavor due to the variety of conditions an AV has to operate in and the complexity of their system implementation. The localization and perception components in an AV take in sensor and map information to create a *world model* to capture the environment around the AV. This world model is then passed to the planning module to create a safe trajectory based on its objective. The perception components based on cameras and LiDARs are increasingly implemented using ML models for 2D and 3D object detection.

It is hard to reason about safety requirements for ML based perception as it is unclear whether (and how) an inaccurate perception would violate a top level safety goal. In practice, requirements of different AV components are driven by subject-matter experts and is largely based on experience. These requirements, moreover, are set conservatively and are common across different driving conditions and Operational Design Domains (ODDs). For example, the localization component should be relatively more accurate on a busy intersection compared to a sparse rural road. Similarly, a perception component should have both high recall and precision on a highway, but could do with just high recall in a pedestrian zone. In an ideal situation, one would like to perceive everything around the vehicle with best possible accuracy using many high resolution (e.g., 24MP) cameras operating at high frame rates (e.g., 120FPS) and employ multiple high accuracy, complex DNN models. Since AVs run on resource constrained platforms, system designers make trade-offs and design a system that is accurate enough (using 2-8MP cameras, 30FPS, and optimized/quantized DNN models with slightly less accuracy, for example). Such a solution based on generic requirements may lead to a system that is less safe for certain conditions where highly accurate perception is required in some areas around the AV (e.g., a fast approaching object from the side at an intersection may need enhanced tracking).

Hardware-in-the-loop (HIL) and software-in-the-loop (SIL) simulations offer effective end-to-end testing methodology for AV systems. HIL testing uses the automotive hardware, sensors, and possibly actuators for system verification and validation. Software-in-the-loop (SIL) simulations are used during the design phase as well as unit and integration testing, where the inputs to the unit or component are either auto-generated or hand crafted to simulate validity with different input parameters.

To enable a AV system design that makes better use of the resources for safer driving, we propose a simulation driven approach to compute world model accuracy requirements for safe AV behavior. As this simulation-based approach is computationally expensive, we also describe efficient methods to explore the state space. Our approach provides the following benefits: (i) it allows to derive component level requirements from the top-level system requirements and driving scenarios, enabling traceability, (ii) it facilitates customization of the world-model accuracy requirement for the driving scenario, allowing to compensate inaccuracies in one component via higher precision in another, (iii) it leads to a formal framework for investigating component level failures in integration and system level testing as needed by classical safety standards ISO 26262 [6] and Safety of the Intended Functionality (SOTIF) [7], which is currently missing in top-level driving scenario simulations.

The remainder of the paper is organized as follows. Section 2 describes the related work. Section 3 presents a formalism for safety in driving scenarios and challenges in component-level design and testing for AVs. Section 4 provides the methodology for computing the world model accuracy and its applications to requirements derivation and testing. Section 5 concludes the paper.

2. Related Work

There has been substantial work on top-level scenario based simulation to establish confidence in AV behavior. OpenScenario [2] is a high level language for describing scenarios for simulation. Waymo recently published their study [17] that demonstrates the behavior of their AV on accident scenarios. Several studies further explore scenario manipulation to discover unsafe AV behavior. Li et al. [9] use fuzzing to manipulate scenarios and discover unsafe AV behavior. Similarly, Tuncali et al. [20] develop a simulation based adversarial testing framework. Smart scenario generation to speed search [14] and accelerate rare-event probability evaluation [15] have also been explored. Ghodsi et al. [5] also employed a method to generate adversarial scenarios along with a method to characterize them based on the how hard or easy it might be for the AV to maintain safety. Zhao et al. [23] develop a framework called Suraksha to study the impact of degraded perception on AV safety. Menzel et al. [12] analyze a scenario abstraction to create an approach for the design of vehicle guidance systems following the development process of the ISO 26262 standard. ML whitebox testing [16, 10] to analyze scenarios based on neuron coverage has also been studied.

Given the importance of machine learning in autonomous driving, recent work has extensively focused on robustness of machine learning for safety critical applications. Sina et al. [13] discuss challenges and ideas for extending classical automotive standards to ML safety. Singh et al. [18] explore the impact of automotive system design on ML based perception. The recent UL4600 [21] standard and the ML safety lifecycle [3] provide guidelines towards established safety of ML based perception components, however they rely on a sound requirements engineering methodology to realize such perception components. Similarly, the Safety First For Automated Driving whitepaper [1] recommends using a checklist based approach for developing the specification for the perception related tasks.

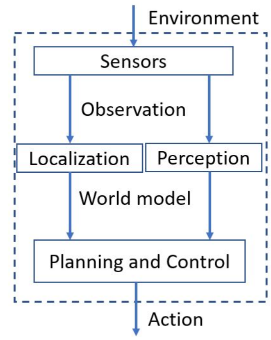

Figure 1. AV Architecture

However it also acknowledges such a checklist is easily outdated. Scaling such an approach for different perception related tasks is challenging. Vogelsang et al. [22] acknowledge the challenges for requirement engineering (RE) for ML- based systems. They claim that RE lifecycle needs to be divided into activities such as elicitation, analysis, specification, validation and verification that requires close coordination between scientists, requirements engineers, legal experts and customers. Caroline Hu et al. [4] propose an approach for specifying requirements for robustness towards small perturbations to inputs for perception component based on ML. This method does not decompose system level requirements, rather uses human performance as a benchmark.

3. Designing and Testing AVs

A modern autonomous vehicle (AV) consists of the following components: a set of sensors to observe the environment, a perception module to detect the static and dynamic objects in the environment, a localization module to estimate the location of the AV on the map, a trajectory prediction module to predict the behavior of the dynamic objects, a planning module to generate a driving trajectory, and a control module to generate control commands which drives the vehicle (including an online collision prediction and avoidance model). For our purposes, we simplify this architecture and let the planning module include the prediction and the control modules. This is illustrated in Figure 1.

3.1. Driving Scenarios

The output of the perception and localization modules is a world model that captures the state of all the static and dynamic objects as well as the ego vehicle. The attributes of an object include its category (or type), shape, location, velocity, acceleration, and other information needed for accurate and comfortable driving. We denote the *world model state* at a given instance by σ.

An ODD specifies the constraints for the safety analysis for the AV. These constraints (e.g. an area or a road segment with limits on the speeds of the vehicles) bound the space and allow for a practical analysis. For a given ODD, a hazard and risk analysis (HARA) is performed using the ISO 26262 [6] and 21448 (SOTIF) standards [8]. For each hazard, a set G of safety goals is derived. For example, in a highway ODD, a hazard is a longitudinal collision due to sudden change in velocity of the ego vehicle. This yields a safety goal that the AV shall prevent unintended longitudinal acceleration/deceleration.

For each safety goal $g \in G$, a set Θ_g of driving scenarios can be created for testing. Each scenario $\theta \in \Theta_g$ includes variations in the environment, number of actors, their starting positions and actions during the course of the scenario. A scenario consists of a map section where the actors and AV are traveling, environmental conditions (e.g. weather and road condition), initial speeds and locations of a set of actors and the AV, a set of maneuvers each actor will take during the execution. We assume that the actors follow an intelligent driver model (IDM) [19]. A scenario θ starts at time 0 and ends at T. We say that the AV is safe in a scenario if AV does not come within a distance r from any other object at any timestep during the scenario. The parameter r depends on the safety goal. In this work, we only consider scenarios that are safe assuming an IDM for AV.

3.2. The Design Challenge

The classical automotive safety standard ISO 26262 starts with an item definition, followed by a hazard analysis and risk assessment. This provides a set of safety goals, which lead to safety requirements. Traditionally, the top level system safety requirements are allocated or decomposed across components.

Such a decomposition is hard for the increasingly complex AV software stacks because (i) allocating or decomposing the requirements to an adaptive planning algorithm is challenging, (ii) the AV system consists of multiple components such as sensors, perception models (e.g., obstacle, lane, intersection, road-sign detection) and localization that work together to create a world model where each component is intrinsically inaccurate, and (iii) the perception component in AVs is based on ML, introducing new, less-understood failure modes due to its black-box nature, that are not captured in classical automotive standards.

In the absence of a top-down requirement decomposition, a bottom up design approach imposes requirements on individual sensor, localization, and perception components. For example, the sensor component imposes a requirement bounding sensor noise, and the perception component imposes an accuracy requirement on object detection. Such generic constraints result in over-engineering certain components and restrict engineers to provision a static system, i.e., deploying same amount of resources independent of the world model and the driving scenario. This misses the opportunity for adaptive design: for example, an AV could boost perception ability for cross-traffic at intersections and allow to track fast-approaching vehicle that may violate a traffic signal and collide. While the AV is not at fault in this example, the overall safety can increase with adaptive perception.

To tackle this problem, our proposal uses top-level scenario simulation to derive requirements for localization and perception. Intuitively, we bound the world model error in the state observation in order to obtain component-level requirements in a way that the AV does not violate the safety goal.

3.3. The Testing Challenge

The general verification problem for a safety goal g is to ensure that the AV is safe for all driving scenarios Θ_g. This is intractable as the number of possible scenarios is theoretically unbounded: two scenarios could differ in the speed, the location, or the size of an object, or the total number of objects. Instead, automotive safety standards [6, 8] require rigorous hierarchical validation through unit, integration, and system level tests. Unit tests ensure proper functioning of the individual units through structural coverage. The integration tests ensure that the static and dynamic aspects of the interaction between the units is well tested. The system tests validate the system behavior. ISO26262 mandates that multiple objectives are met at each level to minimize safety risk. For example, it is highly recommended that system level tests consist of an equivalence class analysis and fault injection tests.

Today, the practice of limiting ourselves to top-level scenario based simulation falls short of the expectations of classical safety standards. We show how our top-down requirement derivation also helps to perform more systematic tests.

4. Simulation Driven Requirement Derivation

We now describe our methodology to characterize safe world model error and create component level requirements using top-level scenario based simulation.

4.1. Approach and formulation

We leverage the driving scenario simulation framework to decompose the safety requirements down the AV stack to derive the component-level requirements. Our methodology is summarized in Figure 2. For a scenario θ, the goal is to obtain Γ_θ, a set of sequences of perceived world model error values that will not result in a safety violation. We refer to the sequence of errors as $\hat{\gamma} \in \Gamma_\theta$ and the errors in each timestep as $\gamma_0 ... \gamma_T$ (subscript refers to the time-step) where the scenario θ is implicit for sake of notation, and T is the length of the scenario θ.

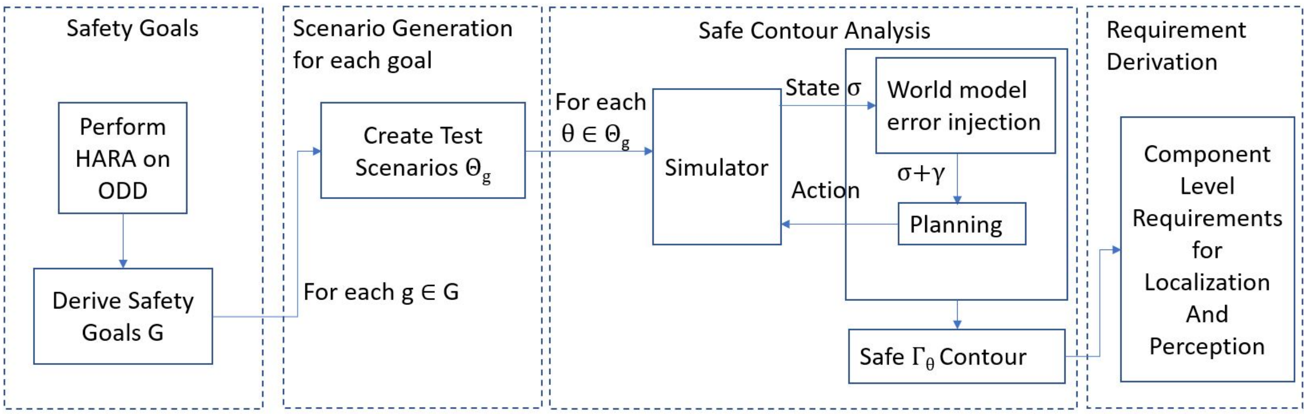

Figure 2. Methodology for computing safe contour of world model error and deriving component level requirements

At each timestep t_i in the scenario θ, an error component γ_i is added to the world model state σ_i and given as input to the planning algorithm which then generates actuator values (actions). The actuation changes the state of the AV in the simulator for the next time-step t_{i+1}. The simulator also updates the states of all other actors based on the scenario description and passes the state σ_{t+1} to the AV for t_{i+1}.

Intuitively, Γ_θ specifies the requirement for (and can also be used to test) the combined perception and localization task, i.e., if the observed error is less than the values in Γ_θ during a scenario θ the AV will be safe. This is discussed in more detail in Section 4.4. *Our goal is to obtain the contour that encompasses Γ_θ, i.e., all the safe sequences of world model error $\hat{\gamma}$ for the scenario θ.*

4.2. Computing the safe contour

Uniformly sampling the discretized N-dimensional space is an approach that remains practical only for small N and relatively large discretization granularity. A fully stochastic approach would randomly sample the entire space with the desired granularity of discretization for each dimension. This approach (shown in Figure 3) can identify the set of points (one point per $\hat{\gamma}$) where the AV remains safe and the set where it does not, which can be used to define an approximate contour. This approach can be bounded by first finding the maximum γ_i for each dimension by keeping all other $\gamma_j = 0$ $(i \neq j)$.

Our approach produces conservative requirements for a world model that leads to safe AV behavior. We can further refine the requirements by examining the range of world model state between the established safe state Γ_θ and states that are unsafe by using fault injection to directly perturb the world model state. The world model state is multi-dimensional with many objects, each of which have multiple parameters such as position and velocity. Accordingly, the search of the world model state to find the unsafe states may be complex because the parameters may be interde-

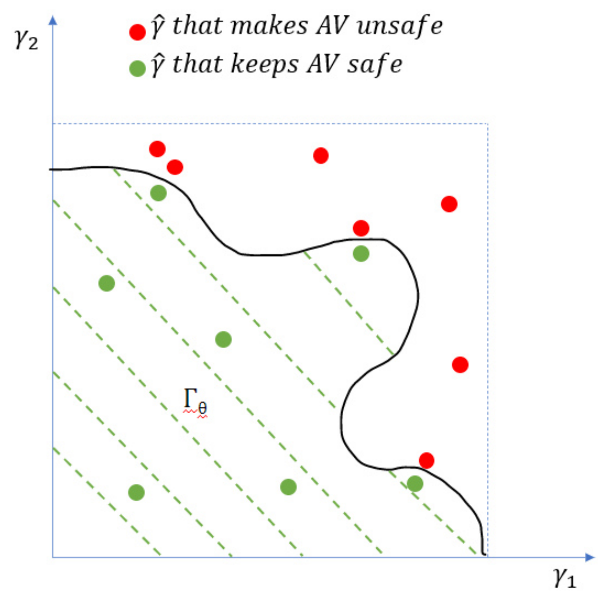

Figure 3. An example visualization of the N-dimensional space (N=2) for Γ_θ where each green colored dot represents a sequence of world model errors $\hat{\gamma}$ that does not impact safety for a given scenario. The two axes represent the world model error in each of two timesteps (which represent the dimensions in this example).

pendent with regard to safety. A straightforward approach for fault injection is to perturb a single parameter at a time while hold other parameters constant. For example, fault injection could be performed using hazard and operability study (HAZOP) based guidewords such as more (qualitative increase), less (qualitative decrease), early (relative arrival/departure), late (relative arrival/departure), no or not (negation), other than (substitution). As the world model state space is explored, the specific boundary between safe and unsafe states yields the more precise requirements for world model accuracy.

4.3. Efficient error space exploration

We now look at techniques to efficiently explore the world model error space.

Constrained error: One natural way to limit the error space is to obey the physical and realistic constraints. For example, limit an error in a non-AV vehicle's location from placing it in sky or out of the road, or a lane can only be straight and and shift but still be on the road. Error that do not follow such constraints can easily be detected and corrective actions can be taken, e.g., fall back to a back-up system or alert the human driver.

Equivalence class analysis based on a safety score: Change (or error) in some attributes of the world model can have similar effect on the safety of the AV while driving in a specific scenario. Such errors can be considered equivalent. With the use of heuristics that predict the safety score, we can skip the simulation if the safety metric is expected to be similar to a simulation that has been already performed.

Limit to obstacles that are most likely to collide: Given that we know the scenario, we can determine the obstacles that are close and are likely to collide with the AV if the AV misbehaves. For example, relative velocity and distance can be used to determine whether an accident is possible if the AV accelerates with some constraints. Methods that estimate Time To Collision (TTC) can be leveraged to identify the objects to define the error space [11]. These objects can be ahead, behind, or on the sides of the AV. This object pruning (or selection) method can significantly reduce the total explored error space. This approach has been consider in the Suraksha framework [23].

Abstract models for fast exploration: Using a detailed simulator that models all the world objects in a photo-realistic manner to capture realistic sensor data and a detailed production-quality AV may be prohibitively slow for deriving the requirements (and eventually tests based on them). Using a setup that can simulate a scenario that takes about 10 seconds in wall-clock time to just a few milliseconds (offering 100-1000x speedup) can make the exploration feasible. Such a setup can be possible with the use of a fast simulator that only models kinematics and transfers the world model directly to the AV by skipping the following steps: sensor data extraction in the simulator, transferring the data to the AV, processing the sensor data in the AV, perception and localization modules in the AV, and actuation step in the AV.

Gradient-based requirement propagation: Attributes of the world model objects include continuous variables. For example, the distance of an object from the ego vehicle or the velocity of that object are continuous variables. Thus, the N-dimensional error space is largely continuous. Each point in the space can be associated with a continuous safety score that represents the relative safety of that point. This view of the world model space allows a gradient-based search that potentially finds the boundary between safe and unsafe points. The procedure for a gradient-based search first requires the defining of a safety score as a scoring function of the N dimensions of the world model space. For example, the distance from the ego vehicle to the nearest object is a possible scoring function. The gradient-based search is conducted by evaluating points in a chosen dimension until an unsafe point is discovered. Thereafter, each dimension is explored until a safe point is discovered. To account for dependencies among the dimensions, several rounds of evaluating each dimension may be required.

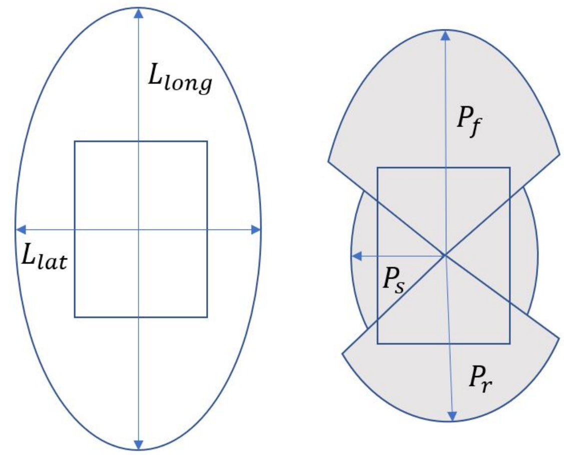

Figure 4. Parametrized requirements for localization and perception.

4.4. Deriving requirements and tests

Translating safe contour to requirements: A primary objective for the safe contour computation is to identify the safety tolerance due to error in the localization and perception components. To do this, the world model error can be divided into two parts: (i) a localization error that is parametrized by L_{long} in the longitudinal and by L_{lat} in the lateral direction (ii) a perception error characterized by portions of a circular area with radius P_f in the front of the AV, P_r in the rear, and P_s at the side of the AV. Here, we choose a static definition of front, rear, side (fixed angles with respect to heading direction of the ego vehicle). We can assume that the perception component accurately identifies an object if the bounding box of the object overlaps with the ground truth bounding box with IoU (Intersection over Union) above a certain threshold, and the detected class matches the class of the ground truth object. This is shown in Figure 4.

Requirements traceability and aggregation: The safe contour model provides traceability between driving scenarios and derived requirements for localization and perception components, as every requirement can be traced to a driving scenario simulation resulting in an unsafe state. Since derived requirements may marginally vary across different scenarios, it is desired to aggregate these requirements according to similar driving scenarios. Exploring methods to group scenarios into equivalence classes based on error tolerance is an interesting research direction. Such a method enables specialization to provide better safety. For example, boosting perception for cross-traffic at intersections can make an AV safer as cross-traffic violating signals or road-signs are among the common driver mistakes that lead to unsafe situations.

Component-level testing methodology: Derived component-level requirements provide a natural decomposition from driving scenarios to component level tests. These tests can be used in HiL and SiL simulations to ensure that the localization and perception components satisfy these tests. Precise requirements also allow to easily inject low-level faults in localization and perception components to validate our safe contour analysis, i.e. if a derived requirement is not satisfied, it results in world model error going outside the safe contour, entering the unsafe territory.

Depending upon the methods used in efficient error space exploration, the derived requirements might make abstract assumptions about the system functionality. Accurate testing shall ascertain the validity of these assumptions, and help diagnose the discrepancies between the deployed system and the assumptions made.

We note that as these simulations for test rely on actual localization and perception components, they are bound to be slower than the safe contour analysis simulation - opening new directions for statistical analyses to perform integration and fault injection tests.

Challenges: Despite the techniques described for efficient state space exploration, several challenges in applying this methodology remain. The parameterized model described in Section 4.4 for perception and localization requirements can include more parameters. Research is needed to understand the effect of using a simpler versus sophisticated parameterization on the system's efficiency. Moreover, the adequacy of the requirements obtained through this simulation-driven approach needs to be tested in real-world settings – primarily to ensure that the artifacts of the simulator do not affect real-world safety.

5. Conclusion

AV systems consist of multiple components for localization and perception to create a world model that is passed to a planning component to create a trajectory. The requirements for different components are based on experience and driven bottom-up, while the safety of AV systems is generally guaranteed using top-level scenario simulations.

We presented a methodology to derive component requirements from top-level driving scenarios. We characterize the permissible world model error as a safe contour, and discuss techniques to efficiently compute the approximate safe contour. We then present ideas to derive component-level requirements from the safe contour and a methodology to test individual components against their requirements. Such a methodology opens new research directions for efficient top-down requirement derivation and modular testing.

References

[1] Aptiv, Audi, Baidu, BMW, Continental, Daimler, FCA, Here, Infineon, Intel, and Volkswagen. Safety first for automated driving. In *Safety First For Automated Driving*, pages 116–132, 2019. 2

[2] ASAM. ASAM OpenSCENARIO 1.0.0. https://www.asam.net/standards/detail/openscenario/, 2020. [Online; accessed 22-Marach-2021]. 2

[3] Rob Ashmore, Radu Calinescu, and Colin Paterson. Assuring the machine learning lifecycle: Desiderata, methods, and challenges. *CoRR*, abs/1905.04223, 2019. 2

[4] Boyue Caroline Hu, Rick Salay, Krzystof Czarnecki, Mona Rahimi, Gehan Selim, and Marsha Checkik. Towards requirements specification for machine-learned perception based on human performance. In *2020 IEEE Seventh International Workshop on Artifical Intelligence for Requirement Engineering (AIRE)*, pages 1–4. IEEE, 2020. 2

[5] Zahra Ghodsi, Siva Kumar Sastry Hari, Iuri Frosio, Timothy Tsai, Alejandro Troccoli, Stephen W. Keckler, Siddharth Garg, and Anima Anandkumar. Generating and characterizing scenarios for safety testing of autonomous vehicles, 2021. 2

[6] International Standards Organization. ISO 26262: Road vehicles - functional safety, parts 1 to 11. In *Road Vehicles - Functional Safety, Second Edition*, 2018-12. 2, 3

[7] International Standards Organization. ISO/PAS 21448: Road vehicles - safety of the intended functionality. In *Road Vehicles - Safety of the intended functionality*, 2019-01. 2

[8] ISO/PAS 21448. https://www.iso.org/standard/70939.html. 3

[9] Guanpeng Li, Yiran Li, Saurabh Jha, Timothy Tsai, Michael B. Sullivan, Siva Kumar Sastry Hari, Zbigniew Kalbarczyk, and Ravishankar K. Iyer. AV-FUZZER: finding safety violations in autonomous driving systems. In Marco Vieira, Henrique Madeira, Nuno Antunes, and Zheng Zheng, editors, *31st IEEE International Symposium on Software Reliability Engineering, ISSRE 2020, Coimbra, Portugal, October 12-15, 2020*, pages 25–36. IEEE, 2020. 2

[10] Lei Ma, Felix Juefei-Xu, Fuyuan Zhang, Jiyuan Sun, Minhui Xue, Bo Li, Chunyang Chen, Ting Su, Li Li, Yang Liu, et al. Deepgauge: Multi-granularity testing criteria for deep learning systems. In *Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering*, 2018. 2

[11] Aashi Manglik, Xinshuo Weng, Eshed Ohn-Bar, and Kris M. Kitani. Forecasting time-to-collision from monocular video: Feasibility, dataset, and challenges, 2020. 5

[12] T. Menzel, G. Bagschik, and M. Maurer. Scenarios for development, test and validation of automated vehicles. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1821–1827, 2018. 2

[13] Sina Mohseni, Mandar Pitale, Vasu Singh, and Zhangyang Wang. Practical solutions for machine learning safety in autonomous vehicles. In *Proceedings of the SafeAI@AAAI 2020, New York City, NY, USA, February 7, 2020*, volume 2560 of *CEUR Workshop Proceedings*, pages 162–169. CEUR-WS.org, 2020. 2

[14] G. E. Mullins, P. G. Stankiewicz, and S. K. Gupta. Automated generation of diverse and challenging scenarios for test and evaluation of autonomous vehicles. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1443–1450, 2017. 2

[15] Matthew O'Kelly, Aman Sinha, Hongseok Namkoong, John Duchi, and Russ Tedrake. Scalable end-to-end autonomous vehicle testing via rare-event simulation. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 9849–9860, Red Hook, NY, USA, 2018. Curran Associates Inc. 2

[16] Kexin Pei, Yinzhi Cao, Junfeng Yang, and Suman Jana. Deepxplore: Automated whitebox testing of deep learning systems. In *Proceedings of the 26th Symposium on Operating Systems Principles*, 2017. 2

[17] John M. Scanlon, Kristofer D. Kusano, Tom Daniel, Christopher Alderson, Alexander Ogle, and Trent Victor. Waymo Simulated Driving Behavior in Reconstructed Fatal Crashes within an Autonomous Vehicle Operating Domain, 2021. 2

[18] Vasu Singh and Mandar Pitale. Impact of automotive system safety design on machine learning based perception systems. In *IEEE Conference on Industrial Cyberphysical Systems, ICPS*, 2021. 2

[19] Martin Treiber, Ansgar Hennecke, and Dirk Helbing. Congested traffic states in empirical observations and microscopic simulations. *Physical Review E*, 62(2):1805–1824, Aug 2000. 3

[20] Cumhur Erkan Tuncali, Georgios Fainekos, Hisahiro Ito, and James Kapinski. Simulation-based adversarial test generation for autonomous vehicles with machine learning components. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, 2018. 2

[21] Underwriter Laboratories. UL4600 voting version. In *ANSI/UL 4600 Standard for Safety for the Evaluation of Autonomous Products*, pages 124–131. UL, 2019. 2

[22] Andreas Vogelsangl and Markus Borg. Requirements engineering for machine learning: Perspectives from data scientists. In *2019 IEEE 27th International Requirement Engineering Conference Workshops (REW)*, pages 1–7. IEEE, 2019. 2

[23] Hengyu Zhao, Siva Kumar Sastry Hari, Timothy Tsai, Michael B. Sullivan, Stephen W. Keckler, and Jishen Zhao. Suraksha: A Quantitative AV Safety Evaluation Framework to Analyze Safety Implications of Perception Design Choices. In *Workshop on Safety and Security of Intelligent Vehicles (SSIV)*, 2021. 2, 5