Causal Models to Support Scenario-Based Testing of ADAS

Robert Maier, Lisa Grabinger, David Urlhart, and Jürgen Mottok, *Senior Member, IEEE*

***Abstract*— In modern vehicles, system complexity and technical capabilities are constantly growing. As a result, manufacturers and regulators are both increasingly challenged to ensure the reliability, safety, and intended behavior of these systems. With current methodologies, it is difficult to address the various interactions between vehicle components and environmental factors. However, model-based engineering offers a solution by allowing to abstract reality and enhancing communication among engineers and stakeholders. Applying this method requires a model format that is machine-processable, human-understandable, and mathematically sound. In addition, the model format needs to support probabilistic reasoning to account for incomplete data and knowledge about a problem domain. We propose structural causal models as a suitable framework for addressing these demands. In this article, we show how to combine data from different sources into an inferable causal model for an advanced driver-assistance system. We then consider the developed causal model for scenario-based testing to illustrate how a model-based approach can improve industrial system development processes. We conclude this paper by discussing the ongoing challenges to our approach and provide pointers for future work.**

***Index Terms*— Causal inference, Bayesian networks, automated driving systems, model-based testing.**

I. Introduction

TESTING Advanced Driver Assistance Systems (ADAS) is a central task in the development of modern vehicles. In the context of autonomous driving, reliable and dependable support systems are essential to provide the necessary technical capabilities to manage the Dynamic Driving Task (DDT). The degree of autonomy of vehicles or the technical capability of support systems respectively is commonly classified based on the six-level decomposition of driving automation given by SAE J3016 [1]. Level zero ($\mathcal{L}_0$) is defined as no automation, meaning that the complete DDT is performed solely by a human driver. On the lowest level with active technical support ($\mathcal{L}_1$), ADAS primarily focuses on warning a human driver and thus provides only temporary aid in critical situations (e.g., lane departure warning system). With increasing system capability, the provided support extends but does not relieve the driver of the DDT up to $\mathcal{L}_3$ (i.e., conditional automation). $\mathcal{L}_4$ and $\mathcal{L}_5$ vehicles are commonly referred to as highly or fully Automated Driving Systems (ADS) which do not require an attentive human driver as a fallback anymore. The difference between $\mathcal{L}_4$ and $\mathcal{L}_5$ capabilities is given by their scope of the Operational Design Domain (ODD). SAE J3016 defines an ODD as:

> "Operating conditions under which a given driving automation system or feature thereof is specifically designed to function, including, but not limited to, environmental, geographical, and time-of-day restrictions, and/or the requisite presence or absence of certain traffic or roadway characteristics." [1, p.17]

Although ADAS capabilities for lower levels of driving automation are not differentiated by a predefined ODD, they are still influenced by its effects. Moreover, for systems with high driving automation ($\mathcal{L}_{3+}$), manufacturers are even required to define a legally binding ODD.

Typically, support systems are developed based on normative regulations like ISO 26262 and established development lifecycles like the therein outlined V-Model [2]. A large part of the ongoing validation and verification effort is grounded in requirement-based testing. Depending on the individual components (e.g., a visual sensor), environmental influences such as rain or fog need to be considered as relevant test parameters. While these two parameters may be obvious from a physical perspective, coming up with all relevant influences is usually difficult.

A promising new approach for system evaluation is the recent effort to test modern vehicles by using so-called scenarios (i.e., a sequence of actions and events in an ODD populated with actors and a System Under Test (SUT)). ISO 21448 specifies how the Safety Of The Intended Functionality (SOTIF) of an ADAS or distinct functions of an ADS can be achieved through the evaluation of scenarios [3]. Scenarios can be executed in real life or with simulators that provide virtual test environments [4], [5], [6], [7].

Although the scenario storyline (e.g., cut-in from the left lane on a highway) can be specified by experts, the unbiased selection of relevant scenario parameters is still under research [8]. From a testing perspective, this set of variables

Manuscript received 16 December 2022; revised 8 May 2023 and 13 July 2023; accepted 7 September 2023. Date of publication 2 October 2023; date of current version 2 February 2024. This work was supported by the Bavarian State Ministry of Economic Affairs, Regional Development and Energy (STMWI) through the granting of the funding Project HolmeS^3 under Grant FKZ: DIK0173/03. The Associate Editor for this article was L. Li. *(Corresponding author: Robert Maier.)*

Robert Maier, Lisa Grabinger, and Jürgen Mottok are with the Software Engineering Laboratory for Safe and Secure Systems (LaS^3), Regensburg University of Applied Sciences, 93053 Regensburg, Germany (e-mail: robert.maier@oth-regensburg.de; lisa.grabinger@oth-regensburg.de; juergen.mottok@oth-regensburg.de).

David Urlhart was with LaS^3, Regensburg University of Applied Sciences, 93053 Regensburg, Germany (e-mail: david.urlhart@gmx.de).

Digital Object Identifier 10.1109/TITS.2023.3317475

defines the context in which a SUT operates in [9], and challenges the intended behaviour of the system within its specified requirements.

In knowledge-driven approaches, the scenario parameters are specified by domain experts. Data-driven methods try to derive parameter distributions and relevant parameter ranges from driving data. Human expertise is often incorporated into the data processing chain in the form of constraints, filtering, or preprocessing activities. This makes expert knowledge the most important source for the definition of an appropriate scenario parameter space. In large part, experts are not guided by a process or by an objective method to quantify the belief that a proposed influencing factor is indeed relevant for a test case.

In all areas and across all levels of detail, there is one common principle: causality. Specifying cause and effect is not only the engine of science but also an essential concept to gain insights into complex systems. In the context of functional safety and the SOTIF, causality is an inherent principle followed during system design and assessment. Similarly, one may ask why the derivation of parameters of interest for testing activities should not also be based on causality—motivating the following research questions:

RQ1 Can an inferable, probabilistic, graphical model be created for a simplified ADAS that includes different levels of abstraction?

RQ2 What are practical aspects when creating and applying such a model?

RQ3 How can these models be used to support scenario-based testing?

In this article, we propose to use a causal model [10] to combine different knowledge sources about a SUT. The causal model then guides the specification of test parameters for the scenario-based testing of an ADAS—a task that can hardly be solved by existing approaches. This model-based approach is discussed by taking a simplified Advanced Emergency Braking System (AEBS) as an example: we show how to build a causal model for an AEBS based on physical considerations, how the modularity of causal models allows linking different sets of concerns, and that model predictions provide realistic results. With that, this article extends previous work on causal model-based engineering [11] by:

- *a running example*: demonstration of a practical development process based on causal models by using a simplified AEBS as an example.
- *a concrete causal model*: definition of a rudimentary causal model including ODD factors for a simplified SAE $\mathcal{L}_1$ ADAS.
- *a concrete parametrization*: parametrization of a causal model in the absence of suitable observational data with the help of literature, domain knowledge, and artificially generated data.
- *inference results*: translation of technical questions about the modeled system at various stages of product development from natural language into inferable probabilistic queries.
- *simulation results*: validation of the inference results with the CARLA virtual simulation platform.
- *conceptual work*: outline the value of causal models and arising challenges for the integration of this approach into automotive industry development lifecycles.

The article is structured as follows. First, we contextualize our work and briefly introduce causal models. Then, basic concepts of an AEBS, the components that constitute the simplified AEBS for the running example, factors of an associated ODD, and settings for data generation are defined. Next, the development and usage of a causal model are discussed. Finally, challenges for an industrial application as well as ongoing areas of research are outlined. The structure deviates slightly from the traditional outline of an article, e.g., instead of comparing our approach to existing work, we start by contextualizing it. This is due to two reasons: first, to the best of our knowledge, no comparable related work is available; second, we anticipate that this change in structure will benefit readers unfamiliar with causal models and allow them to align the proposed method with real-world applications.

II. Context

This section contextualizes the work presented in this article and provides a short introduction to the theoretical framework and, therefore, the enabling method of the proposed approach—causal models.

A. Related Work

1) Status: Ensuring system safety is a key challenge in every technical development lifecycle. In the automotive industry, there are two key normative regulations (i.e., ISO 26262 [2] and ISO 21448 [12]) that outline several safety activities, but avoid giving actual hands-on solutions. While the functional safety assessment of components can be solely based on requirements, specific scenarios including relevant factors of an ODD must be considered to ensure the SOTIF [8], [13], [14]. The latter is commonly framed as scenario-based testing. An important prerequisite for passing a scenario evaluation is the coverage of the correct intended functionality of a component (see ISO 21448). This intended functionality implicitly outlines the test environment: a metric is needed to evaluate the top-level target while a base scenario needs to be instantiated with a combination of all relevant influences. Common approaches to identify relevant scenarios and their parameters include data-driven techniques [15], [16], [17] and domain expert knowledge [18].

2) Challenge: Modern systems can be considered as distributed networks of various components consisting of specialized hardware and software. As a consequence, the use of Machine Learning (ML)-based subsystems has been increasing [19], creating new challenges for an appropriate safety assessment [20]; in large parts, this is due to their interpretation as black-box elements. A key aspect of any testing endeavor is the identification of a suitable set of test cases [21], [22], [23]. Regardless of the capabilities of the SUT, this requires knowledge about its targeted ODD. The challenge this poses is threefold. First, a suitable framework is needed, capable of modeling an ODD in conjunction with other abstract factors. Second, this model should contain enough knowledge about the SUT to derive test parameters from it; this requires that different sources of knowledge (e.g., domain experts and data) can be combined. Third, white-box algorithms are needed for a traceable and certifiable derivation of test inputs and thus the accountable argumentation and accreditation of safety cases.

3) Basis: Modeling causes and their effects is a common method for investigating the behavior of complex systems. In the context of automotive system safety, the interactions between individual components are usually grounded in causality. Decomposing a system in this way allows us to study the evolution of fault chains triggered by root causes (e.g., via fault or event trees). Although the formalism for individual approaches varies, a common concept is to associate specific events with a probability of occurrence. This makes it possible to calculate probabilities for different system constellations and to derive decisions based on the probability theory.

4) Framework: A general, multi-purpose framework for working with (conditional) probability distributions, are Bayesian Networks (BNs) [24], [25] or causal BNs [10]. They allow human expertise and data to be brought together in a structured way. Since they are agnostic to a specific domain of interest, they also allow modeling abstract concepts such as an ODD [11]. In the context of ADAS and ADS, (causal) BNs are commonly used as part of the implementation of the system itself [26], [27]. However, they can also be employed to investigate a black-box system and reason about it [28].

5) Shortcomings: To the best of the authors' knowledge, no current publication provides a step-by-step example of how scenario-based testing and causal modeling can be combined along the V-model, how expert-knowledge and data can be combined in a structured and comprehensible way throughout safety-driven system development, how different levels of abstraction including factors of an ODD can be simultaneously addressed, or how probabilistic insights from a model translate to tangible and safety-relevant engineering activities.

…in scenario-based testing: Riedmaier et al. [14] developed a taxonomy for scenario-based testing and compare existing strategies and methods to address it. They distinguish between expert-based and data-driven scenario generation approaches and find that a combination of both could compensate for individual drawbacks, yet no method is currently available to do so. Similarly, Zhang et al. [29] reviewed current research on scenario-based testing and approaches to identifying critical scenarios in particular. They highlight that reasoning about influencing factors is challenged by the inherent complexity of the DDT and an open context environment. Current methods either focus on data-driven approaches or standardized, step-by-step procedures (e.g., hazard and risk analysis), but usually do not combine these aspects. Moreover, they either cover one level of abstraction or one goal with respect to the SOTIF (e.g., finding triggering conditions or criticality assessment). The inability of current methods to incorporate a variety of concerns (e.g., combining established safety analysis approaches) prevents an early integration of existing methods along the V-Model. The benefit of such an approach is investigated by Thomas and Groth [30] by linking established methods like fault tree analysis or event trees to causal BNs.

…in simulation: Zhong et al. [7] conducted a literature review focused on the connection between scenario-based testing and high-fidelity simulation. They find that the gap between simulator fidelity and the real world needs to be addressed and expressed properly to inform users about potential limitations. A main challenge thereby is the treatment of environmental conditions, and a test object in particular, as a black box. This inherently impedes root cause analysis of failures, as no explicit knowledge is (or can be) incorporated.

…in verification and assurance: Wood et al. [31] discuss various trends for the verification and validation of highly automated driving systems from an industrial perspective. Depending on the individual level of driving automation, unresolved key challenges like a statistical demonstration of system safety, the elicitation of currently unknown scenarios (especially as a result of a changing ODD), or the validation of machine learning-based components (lacking the ability for a logical decomposition into distinct influencing factors or root causes) arise. They state that to achieve a positive risk balance, approaches covering the whole system development cycle (from design to validation and verification) need to be employed, but most methods can only be applied to support some individual stages. Burton et al. [32] outline that for a consistent safety assurance case, various gaps (semantic, responsibility, and liability gaps) in the system specification need to be addressed. They state that this multidisciplinary issue can be solved by continually minimizing the semantic gap, but they do not provide a framework to express these gaps that allow a technical exploitation along a system's lifecycle or facilitates an accessible knowledge exchange (i.e., transdisciplinary communication [33]) among participants.

…in criticality analysis: Neurohr et al. [9] and Koopmann et al. [34] investigate how critical constellations can be systematically identified using causal methods. Neurohr et al. propose to use causal models to analyze the resulting parameter space efficiently:

> "From a formal point of view, we can imagine a causal relation as a network of phenomena where each connection between phenomena represents a plausible cause-and-effect relationship. Note that one causal relation might explain several criticality phenomena at the same time, leading to a condensation of artifacts" [9, p. 7].

This condensation and a subsequent evaluation can be employed via the use of causal methods, which is in line with the findings of Zhang et al.:

> "an unknown critical scenario can be attributed to either an unknown scenario factor or an unknown combination of known scenario factors (…)" [29, p. 6]

Therefore, results can be interpreted as relevant test constellations entailed by an associated ODD. A conceptual process flow to systematically derive and interpret criticality phenomena from a causal model is provided by Koopmann et al. These contributions remain short of a practical, automotive-specific example.

…in summary: To advance automotive system safety and scenario-based testing, the development of new methods is crucial. These methods need to (jointly) tackle challenges such as uncertainty modeling, scalability of scenario generation, risk assessment, and safety validation of machine learning models. By addressing these aspects, new methods can enhance the effectiveness, efficiency, and comprehensiveness of safety testing, ensuring the reliable operation of autonomous vehicles in diverse real-world scenarios.

6) Approach: In the context of safety assessment and scenario-based testing, a causal model-based workflow is an emerging application [9], [11]. In this article, a causal model is used to represent influences associated with the intended functionality of an AEBS. Model inference is then used as a way to investigate the resulting test space. In doing so, we show how a causal model can support scenario-based testing, either by delivering relevant parameter instantiations or by providing a causal rationale for prioritizing test configurations. Prioritization is important when planning a test campaign with limited resources (e.g., time or test infrastructure); with the help of the causal model one can focus on so-called corner or edge cases [11], [35] as candidates for unknown critical situations [12].

B. Theoretical Background

A BN consists of a graphical structure in the form of a Directed Acyclic Graph (DAG) and probability distributions associated with the random variables of the model (i.e., the nodes in the graph). In most graph-based notations, an undirected edge indicates the association between variables, while a directed edge indicates the direction of causal influence. In causal models, a directed relationship is interpreted as a causal mechanism, stating that the dependence of two random variables is attributed to this mechanism rather than on an unknown common cause linking them [10], [36].

BNs formalize how to factorize an underlying joint probability distribution $P(\mathbf{X})$ of a modeled parameter space. The key assumption is that only the direct causes of a random variable x_i (i.e., the parent nodes pa_i in the graph) contribute to its local conditional probability distribution. Hence, $P(\mathbf{X})$ can be given as follows:

$$P(\mathbf{X}) = \prod_i P(x_i|pa_i) \tag{1}$$

Another way to link causal relationships between random variables with their respective probabilistic statements is to use Structural Causal Models (SCMs) [10], [36], [37]. The dependencies between random variables are then formalized by so-called structural equations; the conditional probability distribution $P(x_i)$ of a random variable x_i is given by the assignment-like outcome of a functional description of the interaction of the causal parents of x_i. A SCM therefore specifies each causal mechanism that generates an effect based on its causes and entails a joint probability distribution over all variables in the model. Bareinboim et al. [37] define an SCM as:

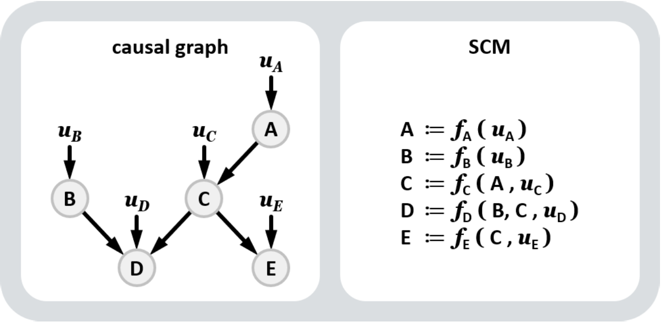

Fig. 1. Representations of a causal model.

Definition 1 (Structural Causal Model (SCM) [37]): A SCM is a 4-tuple $M = \langle \boldsymbol{U}, \boldsymbol{V}, \boldsymbol{F}, P(u)\rangle$ where

1) *$\mathbf{U}$ is a set of background variables (also called exogenous) that are determined by factors outside the model.*
2) *$\mathbf{V} = \{V_1, \ldots, V_n\}$ is a set of endogenous variables that are determined by variables in the model, viz. variables in $\mathbf{U} \cup \mathbf{V}$.*
3) *$\mathbf{F}$ is set of functions $\{f_1, \ldots, f_n\}$ such that each f_i is a mapping from (the respective domains of) $U_i \cup PA_i$ to V_i, where $U_i \subseteq \mathbf{U}$ and $PA_i \subseteq \mathbf{V} \setminus V_i$ and the entire set of $\mathbf{F}$ forms a mapping form $\mathbf{U}$ to $\mathbf{V}$. In other words, f_i assigns a value to the corresponding $V_i \in \mathbf{V}, v_i \leftarrow f_i(pa_i, u_i)$, for $i = 1, \ldots n$.*
4) *$P(u)$ is a probability function defined over the domain of $\mathbf{U}$.*

In general, a causal model consists of a graphical representation (i.e., causal graph), some unconditional probability distributions (i.e., the probabilistic statements assigned to the parentless or exogenous nodes), and a set of causal mechanisms (e.g., via Conditional Probability Tables (CPTs) in discrete BNs or structural equations in SCMs).

A causal graph can be built based on domain knowledge or algorithmically from observational data (i.e., causal discovery [38], [39]). Approaches for parametrizing a causal model (i.e., determining the unconditional probability distributions and causal mechanisms) can also be divided into knowledge [40] and data-driven [25], [41]. In practice, parametrization is done using both approaches simultaneously or iteratively, depending on the availability of data or domain experts for parts of the model.

Note that causal mechanisms are independent of each other (i.e., autonomous [10, Section 1.4.1]) and no assumption about the influence of exogenous nodes $\mathbf{U}$ (noise) is made. This allows the definition of each functional relation specific to the underlying context (e.g., with exogenous additive noise). In practice, purely additive noise models are common but not mandatory and enable the identification of an SCM from data (see also [36, Section 7.1]).

Figure 1 gives a simple example of a causal graph and a corresponding SCM. In this figure and throughout the rest of this article, causal graphs are oriented downward, as this has been shown to enhance comprehensibility [42].

Causal models not only define probabilistic statements and causal interactions (i.e., encode knowledge about a system)

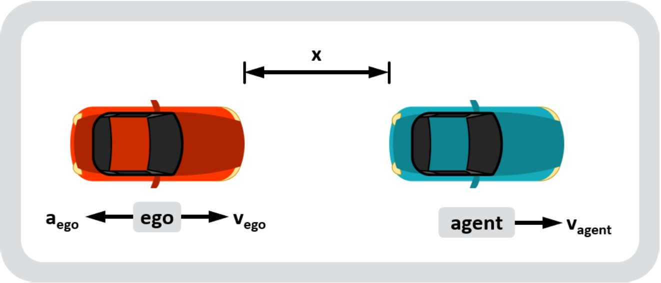

Fig. 2. Setting of the running example.

but also allow to answer probabilistic and causal queries (i.e., generate knowledge about a system). The latter (i.e., inference) can be divided into three categories: associational, interventional, and counterfactual [10]. While associational queries target observations (e.g., $P(x_j|x_i)$), interventional inference allows estimating the consequences of hypothetical actions—again based on observational data [10, p. 32]. These interventions are mathematically formalized as $do(x_i)$ and interpreted graphically as the removal of all ingoing edges of a node x_i. In the case of an SCM, this changes the original structural equation $x_i := f_i(pa_i, u_i)$ to $x_i := do(x_i)$. Counterfactual reasoning allows answering questions such as "what if" and "why". The computation of counterfactual queries typically requires a modification of the causal model concerning the query of interest. Although [10] describes a three-step process for computing counterfactuals, in practice, there are many difficulties (e.g., checking the identifiability of a causal query [43]).

The primary use of an SCMs is to make inferences about the effect of an intervention on a modeled system by quantifying the causal relationships between the variables of the model. This differentiates these models from other methods (e.g., formal methods) by their ability to combine different levels of abstraction and different types of relationships. From a practical perspective, observational data and expert knowledge can be combined and used to ask and answer causal questions.

III. Use Case

This section takes a closer look at the running example—a simplified AEBS. It specifies the intent and elements of the system and describes a simulation setting for data generation and evaluation.

A. Background

The goal of modern ADAS is to support a human driver in potentially critical situations. A prime example of a temporarily acting safety system is given by an AEBS—a type of Forward Vehicle Collision Mitigation Systems (FVCMS). The intended functionality of a FVCMS is specified by ISO 22839:

> "FVCMS mitigate rear-end collisions. By reducing the collision energy, FVCMS reduce the degree of property damage, personal injury, or the likelihood of fatality" [44, p.1].

Therefore, FVCMS must detect [44, sec. 6.1.1] and evaluate objects (i.e., motor vehicles for use on public roads [44, sec. 6.3.1]), process and interpret this data, and activate target functions (e.g., longitudinal vehicle control) if required. An AEBS provides different countermeasures to reduce the severity of an imminent collision. First, a collision warning is issued as soon as a predefined threshold for a proxy measure of the situation criticality such as Time To Collision (TTC) is violated. In this stage, braking as a protective measure is not yet necessary. If no manual action is taken, speed reduction braking is triggered. The intention is to provide an assisted speed reduction and to enable manual emergency braking or an emergency lane change. If a rear-end collision [44, sec. 6.3.2] is unavoidable (e.g., TTC reaches or exceeds a critical threshold), emergency braking is automatically initiated with a minimum specified deceleration [44, sec. 6.1.1]. In the case of an emergency brake, it is not guaranteed that a collision can be avoided completely. Instead, the goal of an AEBS is to reduce the kinetic energy of an equipped vehicle as much as possible and thus minimize potential harm.

AEBS are complex, highly specialized multi-component systems. Since the aim of this article is to show how causal models can be used to investigate parameters for an ADAS evaluation, considering real-life systems is out of scope. This is due to limited resources (e.g., the availability of a test infrastructure or a team of test drivers). Therefore, the focus is on a highly simplified example, which is intended to serve as a proof of concept.

B. Elements

Regardless of a specific scenario or technical implementation of an AEBS, a few generally applicable test parameters can be determined. These include the velocity of the AEBS equipped vehicle (i.e., ego vehicle) v_{ego}, the velocity of a target vehicle (i.e., agent vehicle) v_{agent} ahead, the available ideal deceleration capability of the ego car a_{ideal}, the distance between ego and agent vehicle x, and a fixed Pre-collision Urgency Parameter (PUP). For the running example, TTC is used as PUP due to its widespread use in practice [45]. Based on the relative velocity between ego and agent vehicle v_{rel}, the TTC can be calculated as:

$$TTC = -\frac{x}{v_{rel}} = -\frac{x}{v_{agent} - v_{ego}} \tag{2}$$

Figure 2 visualizes these parameters together with the scenario constellation of the running example. The experimental setup can be interpreted as a typical *car follows leading vehicle* scenario with two participants. The above parameters form the physical base for any emergency braking problem, are independent of each other, and causally contribute to the same effect of interest—whether or not a collision occurred (i.e., is_{col}). A corresponding causal graph can be found in Figure 3. Note that the node TTC refers to a constant value specified at the time of system development (i.e., the PUP threshold), not the value calculated continuously throughout scenario execution.

The causal relations depicted in Figure 3 can be modeled by the equations of motion. This is utilized later on when creating a dataset for the parametrization of the model.

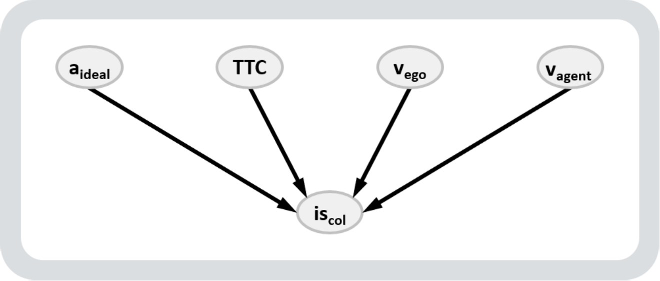

Fig. 3. Baseline causal graph for the running example.

As mentioned above, most ADAS are susceptible to environmental influences via their sensors and actuators. An example of this is given by the actuators of an AEBS: the effective deceleration during braking a_{ego} is a function of the ideal maximum possible deceleration a_{ideal} and the friction coefficient μ between tires and the road surface. The latter depends on the composition of the tire material and whether the road is dry, wet, or icy. Similar considerations can be made for sensor systems and their corresponding algorithms for processing perceptual signals. Object recognition in an AEBS can be realized by a camera system (which continuously monitors the environment in the forward direction) and a neural network for object classification (which processes each camera frame independently). The camera image serves as a direct input to the object classifier. Thus the quality of object recognition is coupled with the quality of the input image. In consequence, the object classifier is subject to environmental factors such as rain, fog, or illumination (i.e., day and night).

In other words, environmental influences link a SUT to an ODD, which makes them relevant for the evaluation of triggering conditions to achieve the SOTIF. While the system itself may be functionally safe (i.e., error-free behavior spanning from data processing to braking), the intended functionality may be compromised in case of heavy rain (e.g., when braking is impossible due to aquaplaning or when visibility is too poor for object detection). Therefore, to ensure the intended functionality of the exemplary AEBS, environmental effects need to be considered. All of these complex interactions work through causal mechanisms that can be modeled by causal models (e.g., causal BNs or SCMs).

C. Simulation

As no suitable observational dataset is available, we generate artificial data for the parametrization of the proposed causal model as well as to evaluate inference results derived from it, the open-source high fidelity simulator CARLA[1] is used. CARLA allows the implementation of a custom AEBS, enables the simulation of environmental influences, and supports scenario-based simulation.

In our custom implementation, emergency braking is automatically initiated when two separate trigger conditions are met at once. The first is the violation of a predefined PUP threshold. Due to its simple implementation and robust results, TTC is used. The second condition is the detection and correct classification of an object in front of the ego car as a vehicle. To simplify the check for fulfilled trigger conditions, the TTC is translated into a minimum distance x_{ttc} to allow a direct comparison with the distance at the first valid recognition of the agent vehicle x_{first}:

$$x_{ttc} = |v_{rel} \cdot \mathrm{TTC}| = |(v_{agent} - v_{ego}) \cdot \mathrm{TTC}| \tag{3}$$

The data processing pipeline of an AEBS can be partitioned into *sense*, *process*, *plan*, and *act* according to the typical decomposition of vehicle control architecture [46]. Figure 4 graphically summarizes explanations of our AEBS implementation presented below:

- *Sense*: Instead of modeling a LIDAR or RADAR for distance measurement, the distance data provided by CARLA is used. Therefore, x is ideal and a calculated TTC is free of jitter. As input to the object classifier, the frames provided by the built-in CARLA RGB camera are used.
- *Process*: Each CARLA RGB camera frame is passed to an off-the-shelf YoloV3 [47] image classifier pretrained on real-life traffic images. Due to its usage in a simulated environment, the classification quality decreases, yet is still sufficient to serve as a proof-of-concept.
- *Plan*: Agent and ego vehicles are controlled by CARLA, which provides the capability to follow a given lane. As this is the only vehicle control requirement for our AEBS apart from emergency braking, no further custom vehicle control modules are needed. To allow the evaluation of braking conditions (i.e., checking the current TTC and ongoing object classification) a custom module is used.
- *Act*: Once all trigger conditions are met, the braking signal is set, manually overriding automatic vehicle control.

IV. Building and Employing the Causal Model

This section describes how different knowledge sources can be used to build a causal model (RQ1) and links it to scenario-based testing (RQ3). Since no real-life dataset is available for the running example, the causal graph is built based on the considerations in the previous chapter. Parametrization is done with the help of literature, domain expert knowledge, and data generated from a custom, causal mechanisms-based Python script and CARLA simulations (RQ2).

A. Building the Causal Graph

As discussed in Section III-B, a causal model used for the assessment of the SOTIF is required to address elements of an associated ODD. In the case of the outlined AEBS, the relevant environmental conditions are the intensity of rain I_{rain}, the intensity of fog I_{fog}, the road surface friction coefficient μ, and whether it is day or night is_{day} (for the sake of simplicity, we refrain from modeling illumination conditions in more detail, neglecting lighting angles, lens effects, blooming, and others).

Rain, illumination, and fog directly affect the image quality captured by the AEBS camera and thus the capability of an

[1] https://carla.org/

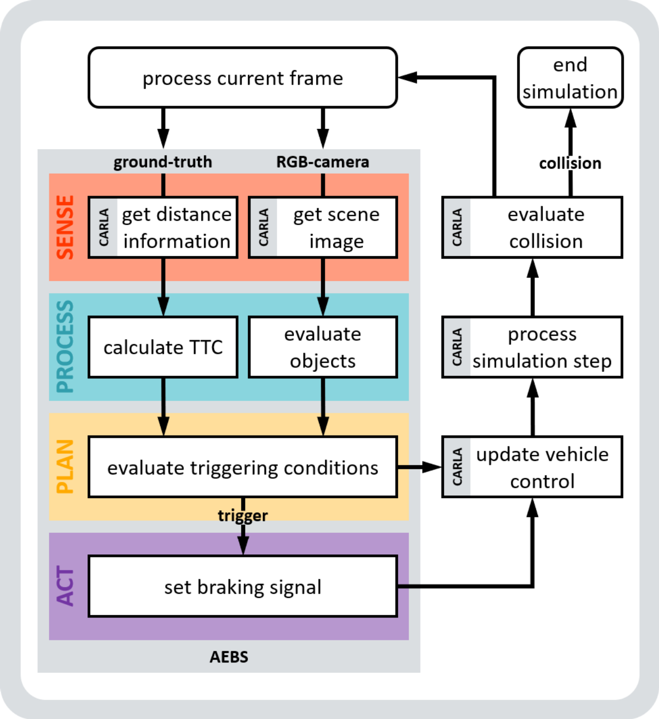

Fig. 4. Data processing pipeline of the running example.

image classifier to recognize objects. A causal expectation is that the distance between the ego and the agent vehicle at the time of the first valid object recognition x_{first} decreases as the environmental conditions become worse. The assumption made throughout this article is that only the environment has an impact on the quality of object recognition. To reduce complexity, hardware influences such as lens effects, decalibration, signal degradation due to image compression, or ML specifics like the quality of the data used for training are ignored.

Friction is modeled to be influenced by rain. As rain intensity increases, friction is expected to decrease. In consequence, the ideal deceleration a_{ideal} decreases to a real deceleration a_{ego}.

In general, attention must be paid when building a model to facilitate a comprehensible, traceable, and justifiable specification of causal relations (expressed via the causal graph and the associated causal mechanisms). For the present work, the simplified assumptions about the problem domain, relevant variables, and their relations were based on expert judgment. If a causal model is to be employed in practice, thorough documentation of the development process, including participating experts, data, algorithms, and assumptions, is required. Only the engineering history in conjunction with the actual model enables sufficient transparency and therefore accountability for derived results.

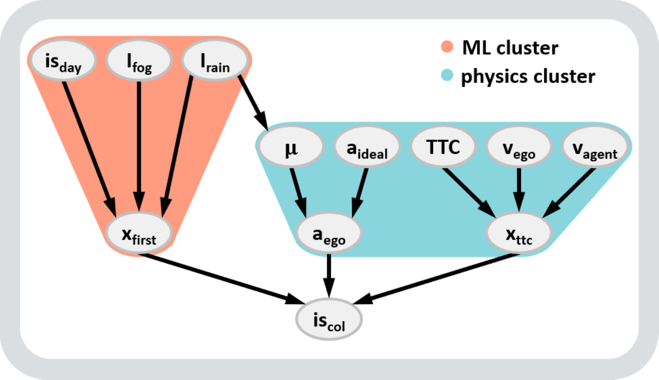

Fig. 5. Full causal graph of the running example.

The resulting full causal graph for the exemplary AEBS is shown in Figure 5. To illustrate the ability of causal models to combine different levels of abstraction or domains, the graph is divided into two clusters (i.e., submodels). The ML-cluster includes influences directly related to the neural network used for object recognition. The physics cluster addresses the driving infrastructure (i.e., via road surface friction) and the physical motion of the vehicles.

Table I summarizes the random variables of the model (i.e., the nodes in the causal graph) and gives a brief description of their intent.

B. Building the Unconditional Probability Distributions

For the specification of realistic distributions for the exogenous nodes (i.e., is_{day}, I_{fog}, I_{rain}, TTC, a_{ideal}, v_{ego}, and v_{agent}), expert knowledge, normative requirements for emergency braking [44], typical values from literature [48], [49], and empirical data are combined. This mixture of methods is valid as the exogenous random variables of the causal model are independent of each other.

The variable is_{day} is defined to be Bernoulli distributed, with one state each for day and night. The probability of $P(day) = 0.5319$ is based on the average time between sunrise and sunset in the city of Berlin (Germany) over the course of one year [50].

The remaining distributions are truncated. This is due to the fact that the support of most variables can be logically constrained to a real-life justified range of values. Above that, with truncation, simple sampling from these distributions does not lead to invalid values (e.g. $v_{ego} = -50.00\,\mathrm{m\,s^{-1}}$), which simplifies the implementation of a data generator. The variables I_{fog} and I_{rain} are modeled as truncated Pareto distributions oriented on literature (e.g., [51]). The actual values (i.e., 0 % to 100 %) are based on the corresponding simulation parameters of CARLA. In real life, a_{ideal} depends on multiple factors such as the type of vehicle (e.g., pickup or small car), its age (e.g., via brake wear), or the road surface. Due to the lack of real-life data, a_{ideal} is estimated to be a truncated normal distribution with a mean value of -5.0 $\mathrm{m\,s^{-2}}$—in line with the required minimum deceleration given by ISO 22839. The mean value of 5.0 s for the PUP TTC is taken from the same standard [44]. The distributions of v_{ego} and v_{agent} are oriented on a traffic study by the German road authority [52]. However, the lower truncation of the distributions differs between ego and agent vehicle. This is because the assumption made for the running example is that the ego vehicle is always moving, while the agent vehicle can be stationary to account not only for moving traffic but

TABLE I
RANDOM VARIABLES OF THE RUNNING EXAMPLE

Random Variable	Unit	Description
is_{day}	{No, Yes}	Whether it is day (i.e., good illumination) or night (i.e., bad illumination)
I_{fog}	*in* %	Fog intensity ranging from no fog to very dense fog
I_{rain}	*in* %	Rain intensity ranging from no rain to heavy rain
μ	-	Friction coefficient between the road surface and ego vehicle
a_{ideal}	*in* $\mathrm{m\,s^{-2}}$	Ideal deceleration of ego vehicle
TTC	*in* s	Critical threshold level of the chosen PUP TTC (static)
v_{ego}	*in* $\mathrm{m\,s^{-1}}$	Absolute velocity of the ego vehicle
v_{agent}	*in* $\mathrm{m\,s^{-1}}$	Absolute velocity of the agent vehicle
x_{first}	*in* m	Distance during the first valid object recognition
a_{ego}	*in* $\mathrm{m\,s^{-2}}$	Actual deceleration of the ego vehicle
x_{ttc}	*in* m	Distance where critical threshold level of chosen PUP TTC is met
is_{col}	{No, Yes}	Whether or not a collision occurred

TABLE II
EXOGENOUS NODES OF THE RUNNING EXAMPLE

Node	Source	Values	Distribution	Parameter
is_{day}	[50]	{No, Yes}	Bernoulli	$p = 0.5139$
I_{fog}	Expert	0.00 to 100.0	Trunc. Pareto	$\alpha = 7.00$
I_{rain}	[51], [53]	0.00 to 100.0	Trunc. Pareto	$\alpha = 7.00$
TTC	[44]	4.00 s to 6.00 s	Trunc. Normal	$\mu = 5.00$ s
a_{ideal}	[44], Expert	$-4.00\,\mathrm{m\,s^{-2}}$ to $-6.00\,\mathrm{m\,s^{-2}}$	Trunc. Normal	$\mu = 5.00\ \mathrm{m\,s^{-2}}$
v_{ego}	[52]	$16.67\,\mathrm{m\,s^{-1}}$ to $55.56\,\mathrm{m\,s^{-1}}$	Trunc. Normal	$\mu = 31.94\ \mathrm{m\,s^{-1}}$
v_{agent}	[52]	$0.00\,\mathrm{m\,s^{-1}}$ to $55.56\,\mathrm{m\,s^{-1}}$	Trunc. Normal	$\mu = 31.94\ \mathrm{m\,s^{-1}}$

also for approaching a stationary agent vehicle (e.g., the end of a traffic jam).

Table II summarizes the resulting distributions together with the knowledge bases used to define them.

C. Defining Causal Mechanisms

Causal mechanisms specify how a conditional distribution is entailed by the distributions of its parent nodes. In the context of this article, they are either defined as deterministic relations via expert knowledge (i.e., for μ, a_{ego}, and x_{ttc}) or learned by algorithms from artificially generated data (i.e., for x_{first} and is_{col}).

According to [49], a typical friction coefficient μ_{ideal} on a dry road (i.e., for $I_{rain} = 0.00\%$) is 0.70. Rain decreases friction down to 0.28 in the worst case (i.e., for $I_{rain} = 100.00\%$). We assume that this influence is nonlinear and includes saturation effects. Accordingly, this mechanism was modeled using a sinusoidal interpolation between the two given extreme values.

The causal mechanism for the real deceleration a_{ego} considers both the ideal friction μ_{ideal} and real friction μ. It is modeled by a linear interpolation starting from the ideal deceleration a_{ideal} as follows:

$$a_{ego} = a_{ideal} \cdot \frac{\mu}{\mu_{ideal}} \tag{4}$$

The causal mechanism for x_{ttc} is only dependent on its parental nodes following Equation 3. The causal mechanism found in the ML-cluster of the full causal graph is more complex: x_{first} is the result of the impaired perception of a scene and its effects on object classification. The corresponding mechanism converts adverse weather conditions (i.e., rain, fog, illumination) and their effects on an ML algorithm, into a metric for triggering a vehicle safety function. This process can be defined as a structural equation:

$$x_{first} := qf(is_{day}, I_{fog}, I_{rain}) + u_{x_{first}} \tag{5}$$

The term $u_{x_{first}}$ is used to adjust for noise in the data, and therefore equation 5 is exemplary modeled as an additive noise model (see also Section II-B). The exact functional relationship between is_{day}, I_{fog}, and I_{rain} might not be specifiable a priori and, in this case, is learned from data. In general, a definition of a structural equation (and in consequence the graphical structure) may in real life be justified from known technical or physical specifications, or, as in this example, defined via expert judgment. It should be noted that each mechanism affects the overall model performance and the reliability and validity of predictions. In practice, an expert-based definition of causal relations and the treatment of noise terms requires special attention when building the causal model. For the running example, the causal mechanism of x_{first} is expected to be expressible by a simple regression model (i.e., third-order polynomial). Due to a lack of real-life data, data generated from a controlled experiment run with the CARLA simulator is used to fit this causal mechanism. The experimental setup follows Figure 2 with a stationary agent vehicle. A scene catalog is created by a variation of the environmental parameters such as rain intensity, fog intensity, and is_{day} as well as the distance between the ego and agent vehicle. For each combination of parameters, a single RGB camera frame is captured and processed by the pre-trained YoloV3 network used for our custom AEBS (see steps *sense* and *process* in Section III-C). If the agent vehicle is correctly detected and classified, the current recognition confidence (i.e., the output of the classifier) along with the configuration of this scenario constellation (i.e., is_{day}, I_{fog}, I_{rain}, and x) is logged. From this log data, the functional relationship from Equation 5, and thus the causal mechanism for x_{first}, is estimated by a 3rd-order polynomial regression on x_{first}. The mean ratio of log data and prediction is 1.0004, indicating a good approximation on average.

Figure 6 shows the intuition of the experiment for three exemplary environmental configurations, based on an existing CARLA preset called *WeatherId*.[2] Figure 7 shows the original data as well as the results of the fitted regressions—split into day and night to allow visual access to the 4-dimensional data.

The causal mechanism of the main node of interest is_{col} can also be formulated ad hoc as a structural equation:

$$is_{col} := f(x_{first}, a_{ego}, x_{ttc}) + u_{is_{col}} \tag{6}$$

Again, in the absence of real-life data, artificial data is created to determine the functional relationship in this equation (this time between x_{first}, a_{ego}, and x_{ttc}). Here, it is sufficient to run a causal mechanism-based data generator written in Python. The corresponding code can be accessed in our GitHub repository.[3] In this script, two moving vehicles are considered. To reduce the overall complexity, it is assumed that the ego and agent vehicles each move at a constant, initial velocity. Once all triggering conditions of the AEBS

[2] https://carla.readthedocs.io/en/stable/carla_settings

[3] https://github.com/othr-las3/aebs-simulator

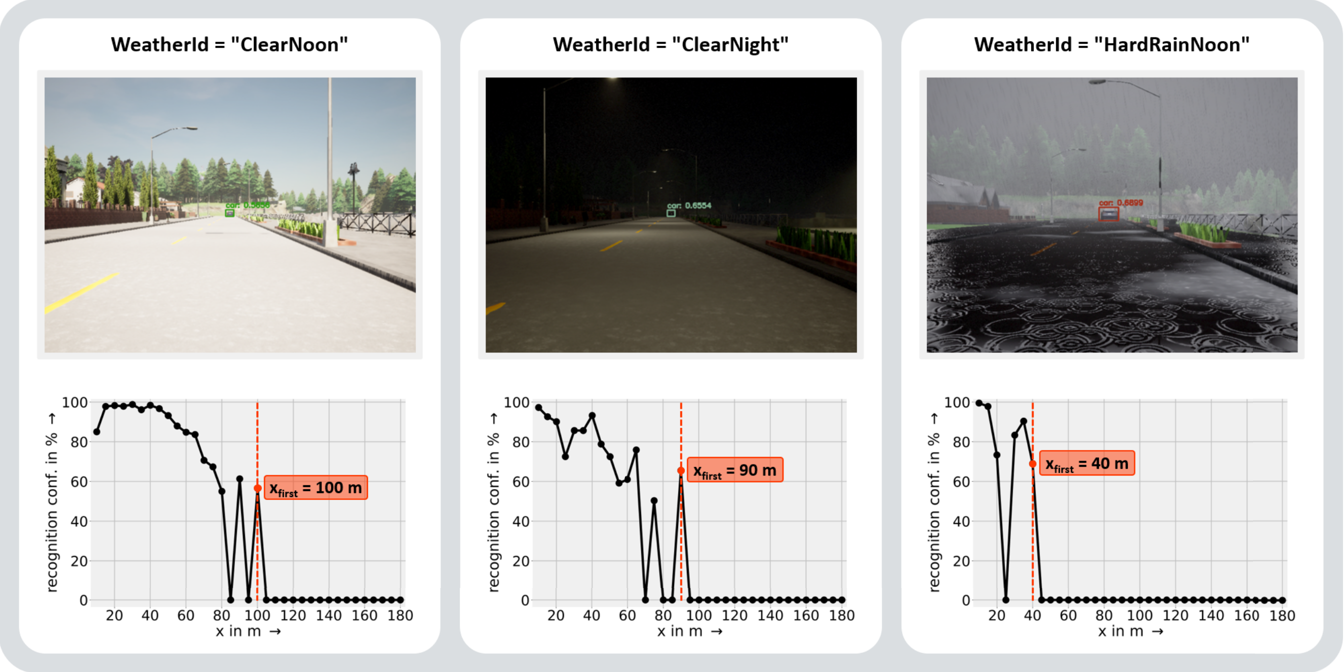

Fig. 6. Experimental setting of the data generation process for the causal mechanism of x_{first}.

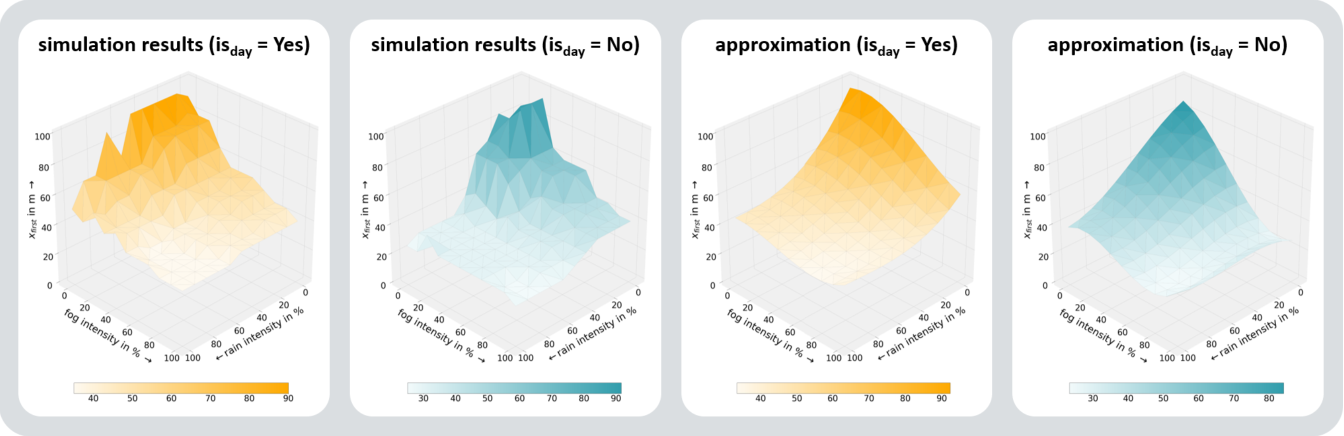

Fig. 7. Regression results for x_{first}.

(see Section III-C) are fulfilled, emergency braking for the ego vehicle is activated, constantly decreasing its velocity. Each data point is generated according to the following procedure:

1) Sample an initial distance x_{init} between the ego and agent vehicle.
2) Sample one value from each exogenous node distribution.
3) Compute the values of x_{first}, a_{ego}, and x_{ttc} based on the respective causal mechanisms.
4) Run a virtual simulation and check whether or not a collision (i.e., the distance between vehicles is lower than 0.00 m) occurs within a predefined duration of 100 s and a time resolution of 0.1 s.
5) Log the initial conditions (e.g., x_{first}, a_{ego}, x_{ttc}, and is_{col}) and the outcome of a testrun (i.e, is_{col}) as a data point.

In the above-outlined data generation process, x_{init} is modeled as a truncated uniform distribution. A constraint for it is that the initial distance must be large enough so that no trigger condition for emergency braking is met with the initial values, but narrow enough so that emergency braking can be activated on average over the duration of a simulation run. The lower limit of the initial clearance can be given by:

$$x_{init\_min} = \min(x_{ttc\_max}, x_{first\_max}) \tag{7}$$

Therefore, the upper limit for the proxy metric x_{ttc} is given by the maximum relative speed and the maximum TTC, whereas the upper limit for the triggering condition x_{first} can be taken from the fitted polynomial of its causal mechanism. As Figure 7 shows, the ideal environmental configuration θ is achieved at day with no rain and no fog. This means that

x_{ttc_max} and x_{first_max} can be computed as:

$$\begin{aligned} x_{ttc\_max} &= (v_{ego\_max} - v_{agent\_min}) \cdot TTC_{max} \\ x_{first\_max} &= \arg\max_{\theta} \; f_{x_{first}}(\theta) \end{aligned} \tag{8}$$

For the running example, the lower bound for x_{init} is calculated to be 380.00 m. The upper limit is empirically determined to be 500.00 m.

D. Implementing the Causal Model

In Section II-B, SCMs and BNs were introduced as established mathematical frameworks for causal models. Implementations of SCMs often resort to Probabilistic Programming Languages (PPLs) like pyro[4] or pymc,[5] which allow a direct representation of structural equations. A major advantage over using discrete BNs is that the estimation of continuous conditional distributions is framed as an optimization problem. Most available PPL libraries do not support causal inference out of the box but can be adjusted in principle. To work with (discrete) BNs, established libraries include open-source packages like pgmpy[6] and pyAgrum[7] and commercial software suits like BayesServer[8] that either take CPTs as inputs or can be used to learn probability distributions or a causal structure algorithmically from data. With regard to causal inference, working with discrete BNs is the most common approach. Therefore, to allow low-effort replication of reported results, we use BNs implemented in pyAgrum version 1.4.1.

The parametrization outlined in the previous Sections IV-B and IV-C is based on continuously-valued data for most nodes in the model (i.e., all nodes apart from the boolean random variables is_{day} and is_{col} have a continuous value range). To estimate the corresponding CPTs, this continuous data needs to be discretized. The specification of the limits, bin sizes, and the discretization strategy (e.g., equal counts or equal ranges) directly affects the quality of the model. On the one hand, a fine-grained approach increases the overall expressiveness of the model but leads to extreme runtime and memory overhead each time probabilistic inference is run. On the other hand, a coarse granularity may oversimplify causal mechanisms, resulting in a loss of insight. An empirically found discretization into 11 equal-width bins yields the best results for the running example. Figure 8 shows the resulting prior probability distributions of the inferable model.

E. Inference of the Causal Model

As stated in Section II-B, counterfactual reasoning faces many challenges in practice. As a result, not every available package supports it. Therefore, in this section, the focus is on associational and interventional inference.

Note that due to the discretization, point estimates (e.g., $v_{ego} = 25.5\,\mathrm{m\,s^{-1}}$) are not possible. Moreover, in the queries below, the indices *max* or *min* refer to the respective discretization interval and its values (i.e., B1 or B11, respectively). In the case of v_{ego}, the minimum value $v_{ego,min} = B_1$ covers velocities in the range of $16.63\,\mathrm{m\,s^{-1}}$ to $20.20\,\mathrm{m\,s^{-1}}$, followed by B2, which covers the interval of $20.20\,\mathrm{m\,s^{-1}}$ to $23.74\,\mathrm{m\,s^{-1}}$. The maximum value $v_{ego,max} = B_{11}$ is defined for the range of $52.02\,\mathrm{m\,s^{-1}}$ to $55.56\,\mathrm{m\,s^{-1}}$.

In the following, two distinct usages of causal model inference are presented—predictive and diagnostic queries. The distinction is made in terms of the actual use case and purely semantically. In this article, collision is the main effect of interest. *Predictive queries* therefore focus on the change of the probability distribution for the node is_{col} based on fixed causes (i.e., outcome prediction). *Diagnostic queries* consider a known state of is_{col} and investigate individual parameter configurations that contribute to the effect of interest (i.e., cause investigation).

1) Predictive Queries: With respect to the AEBS under investigation, some basic associative questions of interest arise. These can be formulated in natural language and as probabilistic queries. For example:

Q1 What is the prior (i.e., unconditional) probability for a collision? $P(is_{col} = Yes)$

Q2 What is the probability of a collision under ideal environmental conditions?
$P(is_{col} = Yes|is_{day} = Yes, I_{rain} = I_{rain,min}, I_{fog} = I_{fog,min})$

Q3 What is the probability of a collision under the worst environmental conditions?
$P(is_{col} = Yes|is_{day} = No, I_{rain} = I_{rain,max}, I_{fog} = I_{fog,max})$

Q4 What is the probability of a collision at the maximum relative velocity?
$P(is_{col} = Yes|v_{ego} = v_{ego,max}, v_{agent} = v_{agent,min})$

Apart from that, interventional questions are also of particular interest. For example:

Q5 What is the estimated effect on a collision if braking capability is maximized? $P(is_{col} = Yes|do(a_{ego} = a_{ego,max}))$

Q6 What is the estimated effect on a collision if the road surface allows ideal friction independent of rainfall? $P(is_{col} = Yes|do(\mu = \mu_{max}))$

Q7 What is the estimated effect on a collision if the vehicle has bad tires? $P(is_{col} = Yes|do(\mu = \mu_{min}))$

Q8 What is the estimated effect on a collision if object recognition is optimized? $P(is_{col} = Yes|do(x_{first} = x_{first,max}))$

Q9 What is the estimated effect on a collision if object recognition is ideal and vehicle traction is poor? $P(is_{col} = Yes|do(x_{first} = x_{first,max}, a_{ego} = a_{ego,min}))$

Table III summarizes the queries above and presents their results. Note that only the shortened conditional statement $is_{day} = Yes, I_{rain,min}, I_{fog,min}$ instead of the entire query $P(is_{col} = Yes|is_{day} = Yes, I_{rain} = I_{rain,min}, I_{fog} = I_{fog,min})$ is given due to space restrictions. To put the individual results in perspective, the ratio between the respective

[4] https://github.com/pyro-ppl/pyro

[5] https://github.com/pymc-devs/pymc

[6] https://github.com/pgmpy/pgmpy

[7] https://agrum.gitlab.io/

[8] https://www.bayesserver.com/

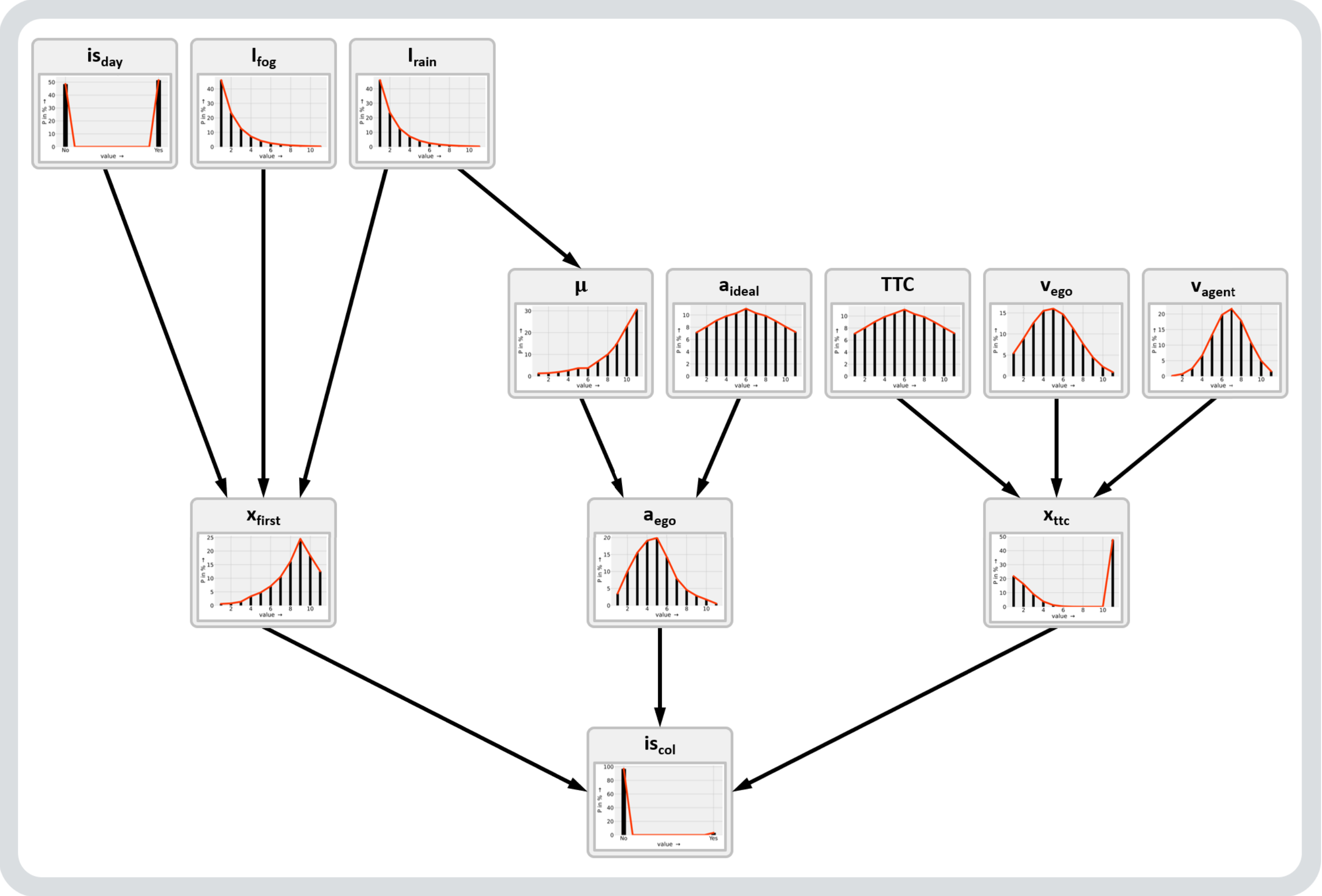

Fig. 8. Fully parametrized and inferable causal model of the running example.

TABLE III

RESULTS OF PREDICTIVE QUERIES

ID	Condition	Result	$\frac{P_{posterior}}{P_{prior}}$
Q 1	-	3.16 %	1.00
Q 2	$is_{day} = Yes, I_{rain,min}, I_{fog,min}$	1.17 %	0.37
Q 3	$is_{day} = No, I_{rain,max}, I_{fog,max}$	19.29 %	6.10
Q 4	$v_{ego,max}, v_{agent,min}$	53.57 %	16.95
Q 5	$do(a_{ego,max})$	1.96 %	0.62
Q 6	$do(\mu_{max})$	2.67 %	0.84
Q 7	$do(\mu_{min})$	19.52 %	6.18
Q 8	$do(x_{first,max})$	2.94 %	0.93
Q 9	$do(x_{first,max}, a_{ego,min})$	50.00 %	15.82

posterior and prior probabilities of a collision is included. Ratios below 1.00 indicate a decreasing collision probability (i.e., a collision becomes less likely), while ratios above 1.00 indicate an increasing collision probability (i.e., a collision becomes more likely) with a given observation or intervention.

Inference results directly yield practical interpretation. For example, the result of query Q4 states that the likelihood of a collision increases by a factor of 16.95 at a maximum relative speed. Likewise, the result of Q2 indicates that a collision is 2.7 times less likely under optimal weather conditions than under the general, unconditional estimate. In particular, the results of the interventional queries Q5 through Q9 may be important during product development or causal model verification. The causal assumption that poor traction (e.g., due to wrong or old tires) leads to an increased braking distance and thus to a higher collision probability, is confirmed by the results of Q6 and Q7. Similarly, Q8 suggests that improving image processing alone might not be sufficient to significantly improve the overall AEBS performance.

2) Diagnostic Queries: In the running example, diagnostic queries amount to the computation of posterior distributions for different nodes based on a (non-)observed collision. The queries can be written as $P(x_i|is_{col} = Yes)$ or $P(x_i|is_{col} = No)$, respectively, with x_i referencing the individual nodes in the causal model. Figure 9 shows the change in posterior distributions depending on whether a crash was observed (central bar) or not (right bar). For each node x_i, the prior unconditional distribution $P(x_i)$ (left bar) is added as a reference.

The results support the causal expectations. For example, if an accident is observed, the probabilities for lower velocities of the agent vehicle increase, while the distribution of v_{ego} shifts toward higher velocities. This is consistent with the general assumption that a higher v_{rel} increases the overall probability of an accident.

Moreover, diagnostic queries can be used to find the most likely explanation (i.e., the most likely parameter configuration) for a given outcome. For example, when considering

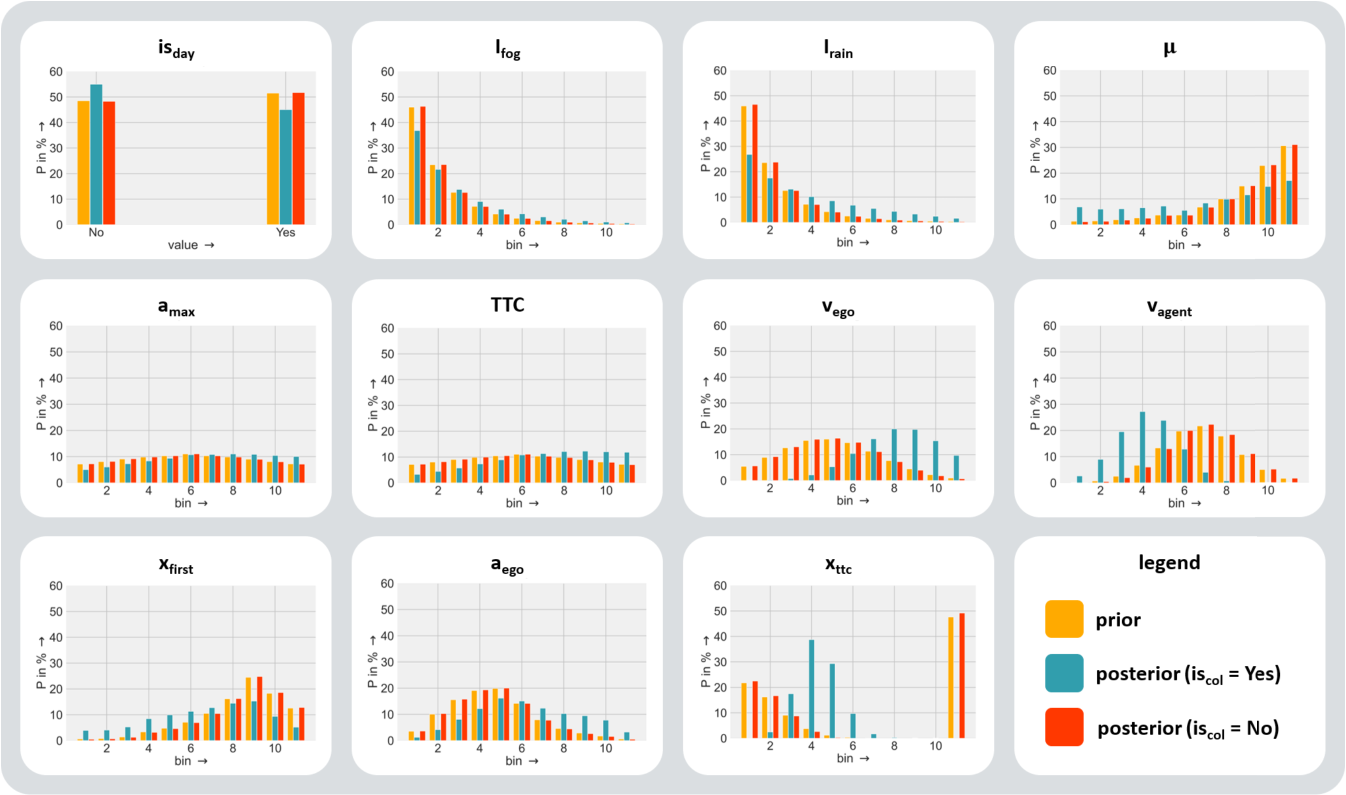

Fig. 9. Probability distributions given a collision (blue), no collision (red), or no condition (yellow).

the influences in the ML-cluster (see Fig. 5), the following questions might be of interest:

Q10 What is the most likely observed configuration of environmental factors (i.e., is_{day}, I_{rain}, and I_{fog}) for optimal object recognition? $\arg\max_{\theta_1} P(x_{first} = x_{first,max}|\theta_1)$

Q11 What is the most likely observed configuration of environmental factors (i.e., is_{day}, I_{rain}, and I_{fog}) for unsuccessful object recognition? $\arg\max_{\theta_2} P(x_{first} = x_{first,min}|\theta_2)$

The computation of these queries is based on [25, p. 26]. Their results support the naive causal expectations. A successful object recognition can be explained by the configuration $\theta_1 = \{is_{day} = Yes, I_{fog} = I_{fog,min}, I_{rain} = I_{rain,min}\}$. Likewise, in case of unsuccessful object recognition, the most likely explanation is $\theta_2 = \{is_{day} = No, I_{fog} = I_{fog,max}, I_{rain} = I_{rain,max}\}$.

F. Checking Plausibility

A common problem in model-based approaches is the fidelity gap between a model and the real world. While in some cases an abstraction of complex processes and interactions is beneficial (e.g., in terms of runtime or communication), it usually affects model validity. In this section, the plausibility of inference results of the causal model is assessed with individual CARLA simulations according to Section III-C. The goal is to build confidence in inference results and thus in the validity of the whole model-based approach.

The exemplary scenario setting is based on the test of functional ability provided by [44]. Accordingly, the scenario storyline is that an (ego) vehicle approaches a slower (agent) vehicle on a highway. The ego vehicle is configured to drive at a velocity of 40.0 $\mathrm{m\,s^{-1}}$; the agent vehicle is set at 18.0 $\mathrm{m\,s^{-1}}$. For the causal model, these values translate into bin B7 for v_{ego} with a range of 37.88 $\mathrm{m\,s^{-1}}$ to 41.42 $\mathrm{m\,s^{-1}}$ and bin B4 for v_{agent} with a range of 15.16 $\mathrm{m\,s^{-1}}$ to 20.21 $\mathrm{m\,s^{-1}}$.

Concerning the configuration above, the following causal queries of interest are evaluated:

Q12 What is the probability of a collision with no rain in the daytime? $P(is_{col} = Yes|v_{ego} = B7, v_{agent} = B4, I_{rain} = I_{rain,min}, is_{day} = Yes) = 12.72\%$

Q13 What is the probability of a collision for heavy rain in the daytime? $P(is_{col} = Yes|v_{ego} = B7, v_{agent} = B4, I_{rain} = I_{rain,max}, is_{day} = Yes) = 92.67\%$

While the query results are added directly to the listing above (Q12 and Q13), the CARLA simulation results are shown in Figure 10. There, the velocity of the ego vehicle v_{ego}, the velocity of the agent vehicle v_{agent}, the bumper-to-bumper distance x, and the deceleration of the first-person vehicle a_{ego} are plotted over the simulation time. The time at which AEBS activation is triggered by a violation of a minimum TTC associated with a successful object recognition is highlighted.

The model estimates of Q12 and Q13 show that, compared to an average weather condition (i.e., in terms of the database and the assumptions made in Section IV-B), heavy rain significantly increases the probability of a collision in the highway scenario above.

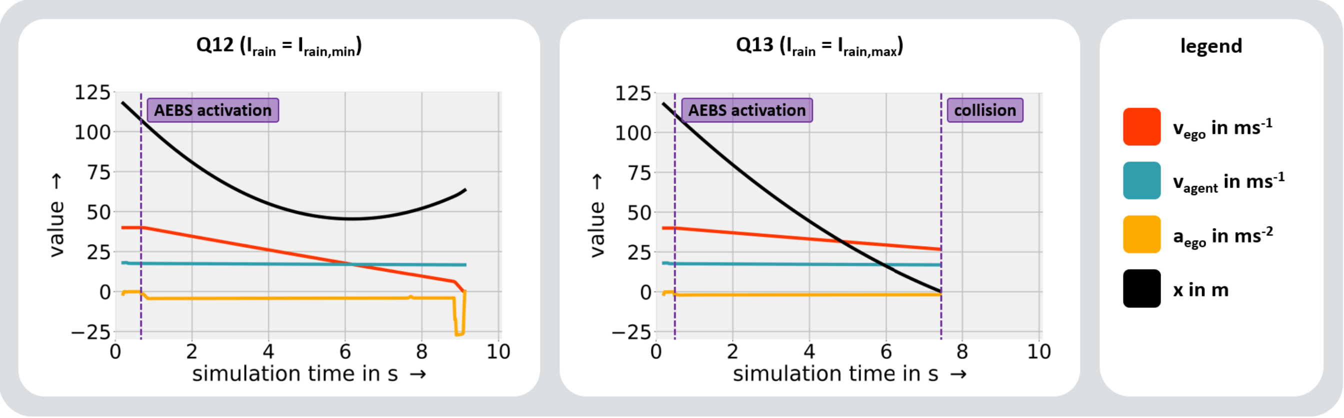

Fig. 10. Simulation results for selected queries.

When the simulation results are analyzed, the same tendency for these configurations (i.e, Q12 and Q13) can be seen. Note that the causal mechanisms derived during model construction do not exactly match with the CARLA implementation, but the principal effects of rain hold true. While for perfect weather conditions (Q12) an accident is avoided (i.e., Figure 10 left), the decreased friction and therefore impaired braking capability of the AEBS (i.e., Figure 10 right) result in a collision.

V. Challenges of the Causal Model Approach

The proposed approach is subject to some challenges. In this section, these challenges are briefly discussed with regard to the running example.

A. Challenges Related to the Sources of Knowledge

Possible Sources: The most critical part of the approach is that of building the causal model. This includes the specification of parameters that need to be considered, the definition of their interactions, and the parametrization of the resulting graph. For this, at least one thing is required: datasets or domain experts. Human expertise is often available, but may still pose a challenge—a structured approach is necessary to elicit the available knowledge and form a consensus when opinions differ. Datasets are easier to manage, but often comprehensive matching and publicly available real-life datasets are not accessible, as is the case for the running example in this article.

Source Mixture: In the course of this article, we combined both potential sources: data, and domain knowledge. This includes published theoretical considerations on parameter thresholds, various weather databases, and artificial data generated based on the CARLA simulator and the causal mechanisms of the model. On the one hand, this enables rapid prototyping of the causal model and adjustment of its level of detail at will. On the other hand, this aggravates the challenge of ensuring quality aspects such as adaptation to the use case or appropriate model complexity. Depending on the actual structure of the causal graph and the intended use of the model itself, mixing different knowledge bases might not be justified–for example, when specific hardware aspects of a sensing system of an SUT are modeled. Depending on the granularity or the level of detail, a resulting (part of the) causal model might be implicitly vendor-specific. When using algorithms to learn either the structure or the causal mechanisms of a model, the usual quality aspects regarding data (e.g., accuracy, representativeness, completeness, or diversity) need to be addressed. This may also justify or reject individual, local assumptions of the model (e.g., structure or the influence of noise) and, in consequence, affect the robustness and generalizability of model insights.

B. Challenges Related to the Model Development Lifecycle

1) Intended Use Case: Causal models follow a particular development lifecycle [54, Ch. 10]. In the context of system safety, the predominant guiding rule is the intended use case [11]. It structures which sources of knowledge are relevant and have implications on the actual modeling process and subsequent use of the model.

2) Subgraphs: In the running example of this article, the identification of different subgraphs proved valuable. Moreover, it enabled the usage of independent data for model parametrization. In larger graphs, extending nodes and mechanisms by specifying attributes, metadata, or clusters helps to structure model development. This can improve the development process itself. We can take as an example a graph that is divided into a sense and a processing cluster. Separating the domains allows to parallelize data collection or generation, expert elicitation (including team staffing), and model validation or verification. Note that the individual subgraphs still need to be developed in accordance with the predefined intended use and that the combination of those subgraphs to a main graph of interest needs to be handled with care. In practice, the latter requires joint meetings of the experts involved in model construction to reach a consensus. Aggregating opinions, data, or even partial models can be improved by resorting to specialized tools and methods [40].

3) Validation: Once a causal model has been developed, validation and verification are required. In the validation phase, the model and its underlying assumptions are challenged—the model is checked to see whether it matches its use case. This works through typical methods such as a model walk-through, stakeholder interviews, or a review of intermediate development products (e.g., submodels). Causal models, as used in the present work, are an abstraction of real ADAS or ADS and their expected interaction with a test environment. They can therefore be considered as "virtual proving grounds" for which the known problems (e.g., face validity, credibility, fidelity) apply [55] and which need to be accounted for during model construction. Even if the validation phase shows a good match with the use case, it cannot be guaranteed that all causally relevant influencing factors have been considered.

4) Verification: While validation activities of other disciplines such as software development can be well transferred to causal models, verification - checking the accuracy of the realization - poses particular challenges. Data may be inappropriate for parameterizing or learning a causal model. For example, data may have been collected only during the day, but the assumptions and use case of the model require nighttime scenarios to be considered. In the case of missing or sparse data, probability distributions cannot be determined without bias - human experts are required to "fill in the gaps" - which is subject to subjective judgement [56]. The robustness of model estimates needs to be challenged. This may be done via a sensitivity analysis [24, Sec. 5.7].

5) Special Case Structure: Verification and validation also involve structural considerations. In causal models, the direction of an edge is of great importance as it specifies the inputs and outputs of causal mechanisms. Here, neither algorithms that restrict themselves to learning based on the correlation between variables, expert judgment, nor group consensus guarantee correct (i.e., causal) direction. While the orientation of an edge might be trivial for most cases (e.g., those supported by mathematical equations), it can be unclear for complex parameter configurations (e.g., cognitive processes of a human driver and their impact on the probability of traffic jam).

6) Documentation: For proper validation, integration, and consequent productive use, continuous documentation of the model development process needs to be available. This enables practitioners to support claims for or against the use of a model on an objective basis. Documentation throughout the development lifecycle includes structured management of data, expert opinions, and secondary information. Moreover, a process for defining and managing consistent, computer-processable meta-attributes for model parameters, mechanisms, (sub)models, and development decisions is required. Adequate consideration of both areas is a prerequisite for credible causal model development.

C. Challenges Related to Scenario-Based Testing

1) Basis: The aim of scenario-based testing with regard to ISO 21448 is to uncover and subsequently manage unknown hazardous situations. Note that the overall scenario content (e.g., the storyline) is often predefined. A challenge arises for the appropriate parametrization of a scenario.

2) Prevalent Approach: In a data-driven approach, parameterization is done by defining a proxy metric for the criticality (and thus test relevance) of a scenario [7], [13], [14]. Based on the chosen metric (e.g., TTC), datasets, simulation results, and real-world situations can be evaluated. Once a predefined threshold is reached, the current constellation of parameters is considered a valuable scenario. By varying either the proxy metric or the variety of traffic situations offered, a catalog of scenarios can be created. This converts previously unknown hazard scenarios into known ones that can be used as a test database.

3) Model-Based Approach: The workflow in this article can be interpreted as a model-driven approach to scenario-based testing [11]. By constructing a causal model of the SUT, a parameter space of relevant influences on system safety is implicitly defined. Therefore, model insights can be used to parameterize existing scenarios, as done in Section IV-F. However, a causal model-based approach can only be used to find and prioritize valuable test parameters and their combinations if these are part of the model. The process outlined in [11] suggests using secondary sources of information (e.g., an hazard analysis and risk assessment) in addition to domain experts and data; yet, it does not guarantee model completeness. Moreover, a structured, algorithmic approach to identifying test cases remains to be explored. Similarly, metrics that consider SUT agnostic variables to define a robustness index [57] would be helpful to compare model insights with real-world data and simulation. Note that the proposed method supports scenario-based testing in a structured way by building a "virtual proving ground" based on existing knowledge. A subsequent evaluation of the SUT based on the generated insights of the causal model (e.g., potential triggering condition and its mitigation) is still necessary [12, Clause 12].

D. Implications With Regard to Safety

Causal models based on SCMs or BNs are suitable to encode knowledge from different sources across problem-relevant domains into a single, inferable model [11]. This is a desired property in the context of ISO 21448, as it is a prerequisite to combine the capabilities and test-relevant properties of a SUT with an ODD and/or a scenario. It allows addressing *safety* from various perspectives. Firstly, the model itself may contain variables that are indicators for the performance of the SUT or for criticality phenomena [9]. Queries can then be structured to estimate the magnitude and direction of causal effects on these variables, which consequently may be used to derive safety principles [34] or guide engineering efforts (as shown throughout this work). Secondly, causal models describe known relevant variables and their interactions in a qualitative and quantitative manner. Moreover, exogenous (i.e. unknown, confounding influences) can be included and considered during model inference. A set of causal models, therefore, is implicitly able to document known testing constellations (e.g., scenarios) and allows supporting the SOTIF activities related to the investigation of hazards implied by the *four scenario categories* of ISO 21448 [12, Sec. 4.2.2]. This includes investigating potential triggering conditions in a structured manner. Thirdly, the development process of a causal model itself enables a rapid feedback loop across stakeholders. Having experts from various departments (e.g., functional safety experts, sensor experts, physicists, etc.) simultaneously develop a causal model in an iterative process [40] encourages discussions that incorporate differing points of view. From the authors perspective, this is different from current practices which are often requirements driven and staged (i.e., centralized responsibilities for different aspects of product development) and lack short feedback loops as well as a shared view on potential emerging insufficiencies. As a consequence, safety-critical constellations are expected to be discovered early on if causal models are used as an accompanying methodology throughout the common V-model development lifecycle.

VI. Conclusion

Due to the rapid development of ADAS and ADS, new challenges for ensuring their safety emerge. Causal models propose one way to address these challenges. Their value comes from their versatility and their ability to include data and expert knowledge to form a probabilistic model of a SUT.

Even systems with a low level of driving automation such as AEBS are based on a complex interaction of hardware and software. In this article, such a reasonably simple system is considered as an example (see Section I) for the combination of causal models and scenario-based testing of ADAS and ADS. A causal model is created by generating artificial datasets based on physical considerations, literature, and normative specifications (addressing RQ1). In the context of early-on product development as well as scenario-based testing according to ISO 21448, engineering questions are derived. By converting these questions from natural language into executable probabilistic queries, causally justified insights about the system are generated (addressing RQ2). Those insights are then used to parameterize an existing scenario template. A subsequent simulation of selected test cases in CARLA verified the derived causal insights (addressing RQ3).

While this article gives a proof-of-concept, additional efforts are required to set up a systematic development and test lifecycle for causal models to enable end-to-end use in an industrial setting. Future research within the “*HolmeS*^{3}” project will address process and tooling requirements and establish a fundamental development infrastructure. The intended development activity includes three clusters: application, methodology, and evaluation. In the application cluster, the focus is on providing tools for expert and data-based model creation as well as model inference. This is accompanied by a methodology cluster that structures the overall *model-to-test* lifecycle covering processes for conducting expert elicitation, incorporation and processing of data, and general quality aspects of causal models. Finally, the evaluation cluster aims at the definition, execution, and management of model and scenario databases. Moreover, additional work is required to investigate the appropriateness of the proposed method for complex, real-life systems. Of particular interest is the robustness of model assumptions and the fidelity of model insights compared to an actual verification (e.g., via simulation or a test drive).

The present article shows that a causal model-based approach offers great potential for supporting the development cycle and safety assessment of modern vehicles, but caution is warranted. In other words: *a model-based approach is good but only as good as the model itself.*

Acronyms

ADS Automated Driving Systems; **ADAS** Advanced Driver Assistance Systems; **AEBS** Advanced Emergency Braking System; **BN** Bayesian Network; **FVCMS** Forward Vehicle Collision Mitigation Systems; **TTC** Time To Collision; **ACE** Average Causal Effect; **CPT** Conditional Probability Table; **DAG** Directed Acyclic Graph; **DDT** Dynamic Driving Task; **ML** Machine Learning; **ODD** Operational Design Domain; **PUP** Pre-collision Urgency Parameter; **PPL** Probabilistic Programming Language; **SOTIF** Safety Of The Intended Functionality; **SUT** System Under Test; **SCM** Structural Causal Model

Acknowledgment

The authors would like to thank David Urlhart, who laid the foundation for this article with his master’s thesis Lokalisierung von Testraum-Hotspots für das Szenariobasierte Testen eines Fahrerassistenzsystems auf Basis von Kausalen Modellen (2022).

Moreover, thanks go to Daniel Ebenhöch from e:fs TechHub GmbH, Tilo Linz from imbus AG, and their respective team members for support, valuable insights, and discussions.

References

[1] ISO Central Secretary. *Taxonomy and Definitions for Terms Related to Driving Automation Systems for on-Road Motor Vehicles*, International Organization for Standardization, Geneva, CH, USA, Standard ISO/SAE PAS 22736, 2021.

[2] ISO Central Secretary. *Road Vehicles—Functional Safety*, International Organization for Standardization, Geneva, CH, Standard ISO 26262-2, 2018.

[3] M. Steimle, T. Menzel, and M. Maurer, “Toward a consistent taxonomy for scenario-based development and test approaches for automated vehicles: A proposal for a structuring framework, a basic vocabulary, and its application,” *IEEE Access*, vol. 9, pp. 147828–147854, 2021.

[4] T. Duy Son, A. Bhave, and H. Van der Auweraer, “Simulation-based testing framework for autonomous driving development,” in *Proc. IEEE Int. Conf. Mechatronics (ICM)*, vol. 1, Mar. 2019, pp. 576–583.

[5] Z. Szalay, “Next generation X-in-the-loop validation methodology for automated vehicle systems,” *IEEE Access*, vol. 9, pp. 35616–35632, 2021.

[6] T. Tettamanti, M. Szalai, S. Vass, and V. Tihanyi, “Vehicle-in-the-loop test environment for autonomous driving with microscopic traffic simulation,” in *Proc. IEEE Int. Conf. Veh. Electron. Saf. (ICVES)*, Sep. 2018, pp. 1–6.

[7] Z. Zhong, Y. Tang, Y. Zhou, V. D. O. Neves, Y. Liu, and B. Ray, “A survey on scenario-based testing for automated driving systems in high-fidelity simulation,” 2021, *arXiv:2112.00964*.

[8] D. Nalic, T. Mihalj, A. Eichberger, M. Bäumler, and M. Lehmann, “Scenario based testing of automated driving systems: A literature survey,” in *Proc. FISITA Web Congr.*, Sep. 2021, pp. 1–10.

[9] 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, pp. 18016-18041, 2021.

[10] J. Pearl, *Causality: Models, Reasoning and Inference*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 2009.

[11] R. Maier, L. Grabinger, D. Urlhart, and J. Mottok, “Towards causal model-based engineering in automotive system safety,” in *Model-Based Safety and Assessment*, C. Seguin, M. Zeller, and T. Prosvirnova, Eds. Cham, Switzerland: Springer, 2022, pp. 116–129.

[12] ISO Central Secretary. *Road Vehicles—Safety of the Intended Functionality*, International Organization for Standardization, Geneva, CH, USA, Standard ISO 21448, 2022.

[13] C. Neurohr, L. Westhofen, T. Henning, T. de Graaff, E. Mohlmann, and E. Bode, "Fundamental considerations around scenario-based testing for automated driving," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Oct. 2020, pp. 121–127.

[14] S. Riedmaier, T. Ponn, D. Ludwig, B. Schick, and F. Diermeyer, "Survey on scenario-based safety assessment of automated vehicles," *IEEE Access*, vol. 8, pp. 87456–87477, 2020.

[15] E. de Gelder et al., "Scenario parameter generation method and scenario representativeness metric for scenario-based assessment of automated vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 10, pp. 18794-18807, Oct. 2022.

[16] W. M. D. Chia, S. L. Keoh, C. Goh, and C. Johnson, "Risk assessment methodologies for autonomous driving: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 10, pp. 16923–16939, Oct. 2022.

[17] P. Karle, M. Geisslinger, J. Betz, and M. Lienkamp, "Scenario understanding and motion prediction for autonomous vehicles—Review and comparison," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 10, pp. 16962–16982, Oct. 2022.

[18] M. L. Cummings and B. Bauchwitz, "Safety implications of variability in autonomous driving assist alerting," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 12039–12049, Aug. 2022.

[19] S. Kuutti, R. Bowden, Y. Jin, P. Barber, and S. Fallah, "A survey of deep learning applications to autonomous vehicle control," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 2, pp. 712–733, Feb. 2021.

[20] A. Eskandarian, C. Wu, and C. Sun, "Research advances and challenges of autonomous and connected ground vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 2, pp. 683–711, Feb. 2021.

[21] X. Hu, B. Zhu, D. Tan, N. Zhang, and Z. Wang, "Test scenario generation method for autonomous vehicles based on combinatorial testing and Bayesian network," *Proc. Inst. Mech. Eng., D, J. Automobile Eng.*, Sep. 2022, Art. no. 095440702211255.

[22] H. Araujo, M. R. Mousavi, and M. Varshosaz, "Testing, validation, and verification of robotic and autonomous systems: A systematic review," *ACM Trans. Softw. Eng. Methodol.*, vol. 32, no. 2, pp. 1–61, May 2022.

[23] W. Ding, C. Xu, M. Arief, H. Lin, B. Li, and D. Zhao, "A survey on safety-critical driving scenario generation—A methodological perspective," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 7, pp. 6971–6988, Jul. 2023.

[24] T. Nielsen and F. Jensen, *Bayesian Networks and Decision Graphs* (Information Science and Statistics). New York, NY, USA: Springer, 2013.

[25] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques* (Adaptive Computation and Machine Learning). Cambridge, MA, USA: MIT Press, 2009.

[26] D. Xu, H. Zhao, F. Guillemard, S. Geronimi, and F. Aioun, "Aware of scene vehicles—Probabilistic modeling of car-following behaviors in real-world traffic," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 6, pp. 2136–2148, Jun. 2019.

[27] K. Pichler, S. Haindl, D. Reischl, and M. Trinkl, "Autonomous highway pilot using Bayesian networks and hidden Markov models," *Electron. Imag.*, vol. 31, pp. 1–7, Jan. 2019.

[28] Q. Zhao and T. Hastie, "Causal interpretations of black-box models," *J. Bus. Econ. Statist.*, vol. 39, no. 1, pp. 272–281, Jul. 2019.

[29] X. Zhang et al., "Finding critical scenarios for automated driving systems: A systematic mapping study," *IEEE Trans. Softw. Eng.*, vol. 49, no. 3, pp. 991–1026, Mar. 2023.

[30] S. Thomas and K. M. Groth, "Toward a hybrid causal framework for autonomous vehicle safety analysis," *Proc. Inst. Mech. Eng., O, J. Risk Rel.*, vol. 237, no. 2, pp. 367–388, Aug. 2021.

[31] M. Wood et al., "White paper: Safety first for automated driving," Aptiv Services U.S., LLC; AUDI AG; Bayrische Motoren Werke AG; Beijing Baidu Netcom Sci. Technol., Continental Teves AG & Co oHG; Daimler AG; FCA U.S. LLC; HERE Global B. V.; Infineon Technologies AG; Intel; Volkswagen AG, Tech. Rep. 1, 2019. [Online]. Available: https://www.connectedautomateddriving.eu/blog/safety-first-for-automated-driving/

[32] S. Burton, I. Habli, T. Lawton, J. McDermid, P. Morgan, and Z. Porter, "Mind the gaps: Assuring the safety of autonomous systems from an engineering, ethical, and legal perspective," *Artif. Intell.*, vol. 279, Feb. 2020, Art. no. 103201.

[33] N. Wognum, C. Bil, F. Elgh, M. Peruzzini, J. Stjepandic, and W. J. Verhagen, "Transdisciplinary systems engineering: Implications, challenges and research agenda," *Int. J. Agile Syst. Manag.*, vol. 12, no. 1, pp. 58–89, 2019.

[34] T. Koopmann, C. Neurohr, L. Putze, L. Westhofen, R. Gansch, and A. Adee, "Grasping causality for the explanation of criticality for automated driving," 2022, *arXiv:2210.15375*.

[35] F. Heidecker et al., "An application-driven conceptualization of corner cases for perception in highly automated driving," 2021, *arXiv:2103.03678*.

[36] J. Peters, D. Janzing, and B. Scholkopf, *Elements of Causal Inference—Foundations and Learning Algorithms* (Adaptive Computation and Machine Learning Series). Cambridge, MA, USA: MIT Press, 2017.

[37] E. Bareinboim, J. D. Correa, D. Ibeling, and T. Icard, *On Pearl's Hierarchy and the Foundations of Causal Inference*, 1st ed. New York, NY, USA: Association for Computing Machinery, 2022, pp. 507–556.

[38] P. Spirtes, C. Glymour, and R. Scheines, *Causation, Prediction, and Search*, 2nd ed. Cambridge, MA, USA: MIT Press, 2000.

[39] M. J. Vowels, N. C. Camgoz, and R. Bowden, "D'ya like DAGs? A survey on structure learning and causal discovery," *ACM Comput. Surv.*, vol. 55, no. 4, pp. 1–36, Apr. 2023.

[40] E. P. Nyberg et al., "BARD: A structured technique for group elicitation of Bayesian networks to support analytic reasoning," *Risk Anal.*, vol. 42, no. 6, pp. 1155–1178, Jun. 2022.

[41] R. Neapolitan, *Learning Bayesian Networks*. Upper Saddle River, NJ, USA: Prentice-Hall, Jan. 2003.

[42] L. Grabinger, F. Hauser, and J. Mottok, "Accessing the presentation of causal graphs and an application of Gestalt principles with eye tracking," in *Proc. IEEE Int. Conf. Softw. Anal., Evol. Reengineering (SANER)*, Mar. 2022, pp. 1278–1285.

[43] J. Tian and J. Pearl, "A general identification condition for causal effects," in *Proc. 18th Nat. Conf. Artif. Intell.* Menlo Park, CA, USA: AAAI Press, Aug. 2002, pp. 567–573.

[44] ISO/TC 204 Intelligent Transport Systems. *Intelligent Transport Systems—Forward Vehicle Collision Mitigation Systems—Operation, Performance, and Verification Requirements*, International Organization for Standardization, Geneva, CH, USA, Standard ISO 22839, 2013.

[45] G. Jahangirova, A. Stocco, and P. Tonella, "Quality metrics and oracles for autonomous vehicles testing," in *Proc. 14th IEEE Conf. Softw. Test., Verification Validation (ICST)*, Apr. 2021, pp. 194–204.

[46] S. Tang et al., "A survey on automated driving system testing: Landscapes and trends," 2022, *arXiv:2206.05961*.

[47] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.

[48] E. Jones and R. Childers, *Contemporary College Physics 2001*. New York, NY, USA: McGraw-Hill, 2000.

[49] A. Abdi Kordani, O. Rahmani, A. S. A. Nasiri, and S. M. Boroomandrad, "Effect of adverse weather conditions on vehicle braking distance of highways," *Civil Eng. J.*, vol. 4, no. 1, p. 46, Feb. 2018.

[50] (2022). *Sonnenaufgang und Sonnenuntergang in Deutschland*. Accessed: Aug. 29, 2022. [Online]. Available: https://www.laenderdaten.info/Europa/Deutschland/sonnenuntergang.php

[51] D. Demand, T. Blume, and M. Weiler, "Spatio-temporal relevance and controls of preferential flow at the landscape scale," *Hydrol. Earth Syst. Sci.*, vol. 23, pp. 4869–4889, Nov. 2019.

[52] U. Löhe. *Publikationen—Geschwindigkeiten auf Bundesautobahnen in den Jahren 2010 Bis 2014*. Accessed: Aug. 29, 2022. [Online]. Available: https://www.bast.de/DE/Publikationen/Fachveroeffentlichungen/Verkehrstechnik/Downloads/Geschwindigkeiten-BAB-2010-2014.html

[53] (2022). *Wetter-Aufzeichnung Für Almhorst Bei Seelze, Kalenderjahr 2021*. Accessed: Aug. 29, 2022. [Online]. Available: http://www.almsick.com/index.php?option=com_content&view=article&id=110&Itemid=732

[54] K. B. Korb and A. E. Nicholson, *Bayesian Artificial Intelligence*, 2nd ed. Boca Raton, FL, USA: CRC Press, 2010.

[55] R. Dona and B. Ciuffo, "Virtual testing of automated driving systems. A survey on validation methods," *IEEE Access*, vol. 10, pp. 24349–24367, 2022.

[56] A. Tversky and D. Kahneman, "Judgment under uncertainty: Heuristics and biases," *Science*, vol. 185, no. 4157, pp. 1124–1131, Sep. 1974.

[57] F. Re, A. Kriston, D. Broggi, and F. Minarini, "Testing the robustness of commercial lane departure warning systems," *Transp. Res. Rec.*, vol. 2675, pp. 385–400, Aug. 2021.

Robert Maier received the B.Eng. and M.Sc. degrees from the Regensburg University of Applied Sciences, Regensburg, Germany. He is currently pursuing the Ph.D. degree with the Clausthal University of Technology, Clausthal-Zellerfeld, Germany. He is a Research Associate with the Software Engineering Laboratory for Safe and Secure Systems (LaS^3), Regensburg. His research interests include the development of novel approaches for investigating system safety in the context of autonomous driving. In particular, he is interested in probabilistic graphical causal models, the development of new software tools for safety analysis, causal inference from observational data, and the industrialization of causal models.

Lisa Grabinger received the B.Eng. and M.Sc. degrees from the Regensburg University of Applied Sciences, Regensburg, Germany. She is currently pursuing the Ph.D. degree with the University of Regensburg, Germany. During the master's studies, she conducted research with the Software Engineering Laboratory for Safe and Secure Systems (LaS^3), Regensburg, on causal inference for system safety in the context of autonomous driving. She is currently a Research Associate with LaS^3, conducting research on and with eye tracking systems.

David Urlhart received the B.Eng. and M.Sc. degrees from the Regensburg University of Applied Sciences, Regensburg, Germany. During the master's studies, he conducted research with the Software Engineering Laboratory for Safe and Secure Systems (LaS^3), Regensburg, on the development of novel approaches to investigate system safety in the context of autonomous driving. He is currently a Consultant in the German automotive industry, advising clients on software engineering and safety.

Jürgen Mottok (Senior Member, IEEE) received the Dipl.-Ing. and Ph.D. degrees from Friedrich-Alexander University, Erlangen–Nuremberg, Germany. He is currently a Research Professor with the University of Applied Sciences, Regensburg, Germany, and the Scientific Head of the Software Engineering Laboratory for Safe and Secure Systems (LaS^3), Regensburg. His lectures and research projects focus on software engineering, functional safety, IT security, and eye tracking.