

# Space, Time, and Interaction: A Taxonomy of Corner Cases in Trajectory Datasets for Automated Driving

Kevin Rösch<sup>1,\*</sup>, Florian Heidecker<sup>2,\*</sup>, Julian Truetsch<sup>1,\*</sup>, Kamil Kowol<sup>3,\*</sup>,  
Clemens Schicktanz<sup>4,\*</sup>, Maarten Bieshaar<sup>5,\*</sup>, Bernhard Sick<sup>2</sup>, Christoph Stiller<sup>6</sup>

**Abstract**—Trajectory data analysis is an essential component for highly automated driving. Complex models developed with these data predict other road users’ movement and behavior patterns. Based on these predictions — and additional contextual information such as the course of the road, (traffic) rules, and interaction with other road users — the highly automated vehicle (HAV) must be able to reliably and safely perform the task assigned to it, e.g., moving from point A to B. Ideally, the HAV moves safely through its environment, just as we would expect a human driver to do. However, if unusual trajectories occur, so-called trajectory corner cases, a human driver can usually cope well, but an HAV can quickly get into trouble. In the definition of trajectory corner cases, which we provide in this work, we will consider the relevance of unusual trajectories with respect to the task at hand. Based on this, we will also present a taxonomy of different trajectory corner cases. The categorization of corner cases into the taxonomy will be shown with examples and is done by cause and required data sources. To illustrate the complexity between the machine learning (ML) model and the corner case cause, we present a general processing chain underlying the taxonomy.

## I. INTRODUCTION

A vehicle equipped with sensor-based assistance systems is driving toward a sharp corner. Suddenly an oncoming vehicle with increased speed appears and cuts the corner. The car driver reduced his speed as a precaution because this dangerous curve was well known to him, and the driver was able to brake just in time to prevent worse. Situations like this, or situations where road users disregard the right of way, are known to every driver. Therefore HAVs must be able to predict trajectories with high reliability. This includes assessing the behavior of other road users and the interaction between them to achieve action planning that can handle such critical situations in the best possible way.

\* Authors contributed equally.

<sup>1</sup>Kevin Rösch and Julian Truetsch are with Mobile Perception Systems, FZI Research Center for Information Technology, Schönfeldstraße 8, 76131 Karlsruhe, Germany, {kevin.roesch, truetsch}@fzi.de

<sup>2</sup>Florian Heidecker and Bernhard Sick are with Intelligent Embedded Systems, University of Kassel, Wilhelmshöher Allee 73, 34121 Kassel, Germany, {florian.heidecker, bsick}@uni-kassel.de

<sup>3</sup>Kamil Kowol is with School of Mathematics and Natural Sciences, University of Wuppertal, Lise-Meitner-Straße 27-31, Wuppertal, Germany, kowol@uni-wuppertal.de

<sup>4</sup>Clemens Schicktanz is with German Aerospace Center (DLR), Institute of Transportation Systems, Rutherfordstr. 2, 12489 Berlin, Germany, clemens.schicktanz@dlr.de

<sup>5</sup>Maarten Bieshaar is with Robert Bosch GmbH, 31132 Hildesheim, Germany, maarten.bieshaar@de.bosch.com

<sup>6</sup>Christoph Stiller is with Institute of Measurement and Control Systems, Karlsruhe Institute of Technology, Engler-Bunte-Ring 21, 76131 Karlsruhe, Germany, stiller@kit.edu

Corner cases are usually extremely critical but (fortunately) very rare situations, which makes them very important for training and validating ML methods. As described in the example, many drivers experience these critical situations, but it is hard to fully define them due to their high variance in appearance. The authors in [1] and [2] found definitions mainly for the perception tasks, but the domain of action, reaction, and interaction in the context of trajectories in space and time is only marginally addressed by these works. This article examines this domain and specifically trajectories in more detail. The main contributions are:

- Systematization and definition of trajectory corner cases,
- Breakdown for corner case situations for trajectory datasets resulting in 31 categories,
- Application of proposed systematization on selected use-cases for filtering of corner case and their targeted synthetic generation.

In Sec. II, we introduce different terms regarding trajectories and corner cases, from which we derive our definition of trajectory corner cases. Sec. III contains a process pipeline that may be used to classify corner cases for both human road users and HAVs, considering examples related to trajectories. Using the taxonomy explained in Sec. IV this work provides a high-level approach to gain a better understanding of how a system can deal with driving hazards. Sec. V shows the application of the taxonomy on simulated and real data.

## II. DEFINITIONS

Before discussing the definition of corner cases in trajectories, we present the probabilistic and generative trajectory model and how corner cases are defined in the literature. We then merge these two substantive points, trajectory model and corner cases, and present our definition of trajectory corner cases.

### A. Generative Probabilistic Trajectory Model

In the following, we introduce a probabilistic trajectory model which is the basis of our corner case definition. Environments and technical systems are often characterized in terms of states  $x$  [3]. A trajectory is a time series of states in some metric space which additionally possesses an inherent spatial smoothness that directly links two neighboring states  $x_{t-1}$  and  $x_t$  in time [4]. Here, the subscript  $t$  denotes the time dependency. A trajectory displays the dynamics of a system under observation in such a state space. Formally, we define a trajectory of

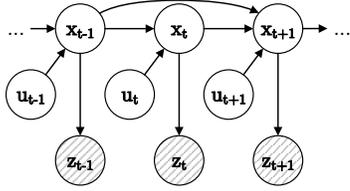


Fig. 1: Probabilistic trajectory model characterized as dynamic Bayes network [3]. The variable  $z_t$  describes the observable, noisy version of the underlying latent state  $x_t$ , which is influenced by a control-input  $u_t$ .

length  $N \in \mathbb{N}$  in a state space as the set  $\mathbf{X} = (\mathbf{x}_t \mid t \in [1, \dots, N])$ . Without loss of generality, we consider trajectories in state spaces comprising 3D-poses  $\mathbf{x}_t \in SE(3)$  in the special Euclidean group consisting of position and the orientation of the system under observation. Note that in many applications in highly automated driving the state space often also comprise single- or multi-order derivatives, e.g., the velocity and acceleration. We model the dynamics of the system under observation probabilistically defined as  $p(\mathbf{x}_{t+1} \mid \mathbf{x}_t, \dots, \mathbf{x}_1, \mathbf{u}_t, \mathcal{C}_d)$ . The evolution of the state  $\mathbf{x}_{t+1}$  may depend on former states  $\mathbf{x}_t, \dots, \mathbf{x}_1$  as well as optional control-inputs  $\mathbf{u}_t$ , e.g., acceleration, braking, or steering input variables. Moreover,  $\mathcal{C}_d$  denotes additional context, e.g., road- and weather-conditions, (traffic) rules, age of driver, or the intentions of the driver and the desired target. In general, we do not have access to the true model underlying the actual trajectory nor do we have knowledge about all context variables. Instead, in practice, we approximate the model as well as the influencing parameters directly in terms of a parametric model or indirectly in terms of a dataset. Furthermore, in most case we cannot observe the true state  $\mathbf{x}_t$  directly, but instead often measure a noise version  $z_t$  of the actual underlying state. Without loss of generality, we assume a simple probabilistic measurement model  $p(z_t \mid \mathbf{x}_t, \mathcal{C}_m)$ , where  $\mathcal{C}_m$  denotes the context relevant for the measurement model. The latter could for example be a potential obstruction. A graphical model of our generative probabilistic trajectory model is depicted in Fig. 1. As before, we do not have access to the exact model governing the measurement generation process. We define an observed trajectory  $\mathbf{Z} = (z_t \mid t \in [1, \dots, N])$ . A trajectory dataset  $\mathcal{D}$  consists of multiple observed trajectories. Note that we do not know the true trajectory model nor the actual context involved. However, we can see observed trajectories as samples from this probabilistic trajectory model.

## B. Corner Case Definition

In the following, we first review different approaches and perspectives to define the term *corner case* before introducing our definition for trajectory corner cases.

### 1) Corner Case Definitions in Literature:

Koopman et al. [5] look at edge and corner cases from the ML testing and validation perspective. Both edge and corner cases are rare cases, e.g., driving or traffic situations, that will occur only occasionally, wherein *rare* means that these cases often occur at large-scale deployed systems,

e.g., full-scale deployed vehicle fleet. An edge case is a situation that requires specific attention during the design time of the system at hand; e.g., these may be sensor values that are out of the predefined bounds or missing values. In contrast, a corner case results from the combination of several “normal” input parameters or situations that coincide simultaneously, thus representing a rare or never considered case or scene [5]. This definition is the basis of many test frameworks, such as DeepXplore [6] or DeepRoad [7]. These methods use differential testing to generate synthetic test cases systematically, also referred to as corner cases. They define corner cases as those in which neural network predictions are indecisive. Hence, the network predicts “I don’t know” [8]. Houben et al. [9] give a general definition of corner cases stating that “inputs that result in unexpected or incorrect behaviour of the AI [artificial intelligence] function are defined as corner cases.” This defines corner cases as situations where the ML model is unable to understand the situation correctly. Others consider mere erroneous or incorrect behavior as corner cases [10]. From the perspective of model training, especially active learning [11], corner cases are the instances, e.g., trajectories that yield the greatest possible learning progress. Following this definition, an instance identified as a corner case and subsequently used for model (re-)training is no longer a corner case. According to Bolte et al. a corner case in the context of automated driving and computer vision is given “if there is a relevant object (class) in a relevant location that a modern automated vehicle cannot predict” [12].

2) *Corner Case in Trajectory Datasets:* Assume we are given a trajectory dataset  $\mathcal{D}$ , e.g., of pedestrian trajectories. Let us imagine, we are given a new trajectory  $\mathbf{Z}_{\text{ped}}$  of a pedestrian intending to cross a street and a vehicle  $\mathbf{Z}_{\text{veh}}$  approaches from far away. Based on the dataset  $\mathcal{D}$  (in the case of humans, these are years of life experience) and other available contextual information, e.g., the environmental circumstances such as zebra crossings and (traffic) rules to be obeyed, we as humans have an expectation of how a pedestrian trajectory should look like. In other words, we have an estimate of the pedestrian’s dynamics and the measurement model (cf. Sec. II-A). However, if a pedestrian stops in the middle of the road to tie her shoes, the pedestrian’s trajectory deviates from the expected trajectory of a pedestrian crossing the road. This suggests that our estimated trajectory model is not explaining the observed trajectory correctly, as something anomalous and unpredictable happened. To what extent a trajectory is unexpected depends strongly on the existing prior knowledge and available context, e.g., in our dataset  $\mathcal{D}$ , we have seen kneeling pedestrians frequently, but not yet in combination with the context of kneeling in the middle of the road. Two common circumstances might lead to an extremely unusual trajectory (since rare). If our task is to train an ML model to predict a pedestrian trajectory as good as possible [4], then such a trajectory can be considered highly relevant and thus a corner case. However, this is rather irrelevant from the perspective of save automated driving as the HAV

is still far away, and there is no interaction between the two. Depending on our task not every anomalous trajectory  $Z_{\text{ped}}$  has to be a corner case. Imagine the vehicle is close by, then due to the interaction of the two road users (given already by the mere presence of the two), such a scene and thus the trajectories of the involved road users experience a completely new assessment. A correct model of the pedestrian’s (future) trajectory is now also highly relevant for avoiding an accident. Another example of such a corner case is a vehicle that passes a red traffic light. The plain trajectory of such a vehicle will not differ from a rule-compliantly driving vehicle. It becomes a corner case only if the context (here the red traffic light) and the disregard of the traffic rules are taken into account. To capture such a corner case, additional information is needed. We conclude that to assess whether a trajectory is a corner case, we must consider the interaction with other objects and agents, i.e., road users, (traffic) rules, and its relevance for the task at hand. Likewise, noisy trajectories with outliers or missing values do not necessarily have to be a corner case. Besides studying relevance, missed measurements, outliers, and missing values can be modeled by an appropriate probabilistic measurement model so that they do not necessarily have to be unexpected nor abnormal. Concluding this into the definition of trajectory corner case within a trajectory dataset  $\mathcal{D}$  as follows:

**Definition Corner Case Trajectory:** *A corner case in a trajectory dataset  $\mathcal{D}$  is a highly relevant but mostly very rare and anomalous trajectory. The relevance of a trajectory is mainly determined by the interaction with other agents (e.g., road users), the surrounding environment, norms and (traffic) rules, and most importantly, by the task at hand.*

### C. Anomalous Trajectories

The terms *outlier*, *anomaly*, and *novelty* are often used interchangeably and are all strongly related to corner cases [1]. In the case of trajectories, we speak of an outlier when one or more measurements deviate so strongly that we assume they were generated by a different underlying process [13]. An anomaly is something that deviates from the standard, average, or expected, whereas an anomalous trajectory refers to the trajectory with local or global differences from most other average trajectories when measured by some similarity metrics [14]. Hence, by definition, an anomalous trajectory is rarely represented within the trajectory dataset at hand. An agglomeration of anomalies as clusters is called novelty [15]. The challenge of detecting anomalous trajectories is referred to as anomalous trajectory detection [14]. Among others, we can distinguish between the following approaches to detect anomalous trajectories: clustering- [16], distance-, density-based [17], and deep-learning-based [18] anomalous trajectory detection.

### D. Interaction

Markkula et al. [19] defines *interaction* as a situation in road traffic in which the behavior of at least two road users

can be interpreted in such a way that both may occupy the same spatial region at the same time. The interaction of road users is the precondition for traffic conflicts, e.g., situations that can potentially lead to an accident [20] and hence a basic building block for determining the relevancy of trajectories. Moreover, modeling the interaction of road users is essential to understand the behavior, e.g., their trajectory, of individual road users in traffic. An example of these are the Gipps or intelligent driver model [21], which describes the “following” behavior of individual vehicles in traffic. There are also numerous interaction models for persons, e.g., the social force model for modeling pedestrian dynamics [22]. Recently, data-driven approaches also explicitly learn interaction patterns to improve prediction, e.g., the social long short-term memory approach [23]. Understanding and modeling the interaction with the environment and between the individual road users is crucial to judge whether a trajectory is anomalous, i.e., contains unexpected interaction patterns, and relevant, i.e., due to a potential conflict, hence, a corner case.

### E. Relevancy

Following our corner case definition, we might be tempted to ask what is relevant and how we can measure the relevance to a given trajectory dataset. Whether a trajectory is relevant strongly depends on the task at hand. Therefore, relevance must be defined on a task-specific basis, making a general definition difficult. In this section, we review different perspectives in the realm of highly automated driving regarding how relevance is defined and measured.

1) *Traffic-Safety Perspective:* Intuition tells us that the relevancy is related to the hazard or risk of a perceived trajectory. Hence, to assess the relevancy of a trajectory and thus identify corner cases, we must inspect the spatial and temporal context of objects in the vicinity of automated vehicles. In other words, the relevancy of trajectories in the highly automated driving domain is correlated with the chance of a crash or near-crash. Therefore, we must also take the interaction of trajectories into account. From the perspective of traffic safety, a trajectory is relevant if it is critical, i.e., the trajectory or multiple trajectories need explicit consideration of potential (crash) mitigation strategies [24]. Many other measures aim to quantify the relevancy of trajectories in terms of criticality or proxy measures such as maximum relative speed, time-to-X, e.g., time-to-break, time-to-collision, time-to-touch [25], the distance of closest encounter [26], time-to-closest encounter [26], and threat-assessment [27]. Kamran et al. define relevant objects in automated driving, which can be transferred to trajectory datasets. According to them, relevant objects are defined by their possible conflict zone with an (automated) ego vehicle. The relevance of trajectories is directly derived from the possible conflict between them [28]. The authors in [14] define a metric to measure the relevance by combining an interaction and anomaly score. However, again which measure is best suited for the detection and quantification of relevant trajectories depends on the precise

objective.

2) *Machine-Learning Perspective*: The prediction is the base of the environment abstraction which is used for decision-making and trajectory planning. The entire processing pipeline consists of many smaller and bigger tasks while every one of them holds their own relevancy. The relevance of individual sensor data, e.g., trajectories and all knowledge abstracted from them are already determined at this point since the selected methodology, model architecture, parameters and hyperparameters for each task have been trained or adjusted accordingly. In self-learning systems such as active learning, the situation is somewhat different. The chosen hyperparameters also determine whether the model classifies the trajectory as relevant for further training or not. Whereas the choice of methodology — choosing between autoencoder and GAN (generative adversarial network) [29] or deciding between a simple detector and a classifier — essentially depends on the task. However, at the end the implemented model architecture determines how the data is processed. For example, the choice of the autoencoder latency directly impacts which information is retained or irretrievably lost. Nevertheless, other design decisions like convolutional layer, fully connected layer, or pooling layer also influence which data is considered relevant. This behavior is also reinforced by the fact that specific feature are weighted more strongly by optimization or model training. For example, if we look at the loss function, which has a direct impact on the model weights during training, the mean squared error loss ensures that outliers are taken into account to a much greater extent than by the Huber loss, which is less sensitive to outliers.

3) *Rule-Compliance*: The goal of rules is to ensure that processes always follow the same pattern and that no participant is put in danger. They provide predictability about the behavior of others. This is particularly essential in road traffic where the violation of traffic rules may result in a dangerous situation. Hence, rule violation within a trajectory is directly coupled with the relevancy of that trajectory; for example, a violation of the right of way may result in dangerous situations. Even the fine catalog offers a possibility to quantify the relevancy of non-rule-compliant trajectories. However, there are indeed situations where an automated vehicle has to perform a “controlled”-rule exception. Imagine an automated vehicle on the street with a solid center stripe waiting behind a parked ambulance due to a rescue mission. In this case, the following vehicles need to enter the oncoming lane, even if the road marking forbids that. This is an easy task for a human driver but a challenging and relevant corner case for the development of HAVs.

### III. PROCESS PIPELINE FOR ROAD USERS

A trajectory corner case taxonomy is helpful to understand critical situations better and consider the reasons leading to the incident. If the taxonomy is used to analyze the misbehavior of an HAV in a specific situation, the first question that arises will be: which part of the system

malfunctioned? If on the other hand, the performance of such a system has to be evaluated more generally in rare and unusual traffic situations, the actual behavior of the other road users is the center of attention. But even then, it is obvious that errors in different parts of the human or machine driver’s information processing system will result in completely different behaviors and trajectories. Therefore, it makes intuitive sense to categorize corner cases by the origin in the driver’s information processing system, whether this is the human brain or the software of an HAV. A unified model of the information processing pipeline for automated vehicles, human drivers, and VRUs (e.g. pedestrians, motorcyclists, and cyclists) leads to a more comprehensive taxonomy while at the same time allowing analysis or classification independent of the level of automation of the road user under consideration.

The task of highly automated driving is typically accomplished by dividing it into sub-problems and corresponding software modules. On the coarsest scale, we can distinguish between perception, decision making, trajectory planning, and control. The *Main Information Processing Pipeline* in Fig. 2 mirrors this partition. Trajectory planning and control are here combined in the *Execution* module. Published system architectures are compatible with this processing pipeline model, cf. [30], [31], [32], [33], although the names of the modules vary.

On this high level of abstraction, the stages of information processing in humans bear many similarities to the ones in highly automated driving software. We use the model of human information processing stages by Wickens [34] as reference for human road users. This very general model is widely used and generally accepted in traffic psychology and traffic safety research [35]. We picked this model over other well-established models from traffic psychology and human factors as many models presume specific cognitive mechanisms that are not applicable to HAVs [36], [37], [38] (e.g. schema activation) in addition to the core components described by Wickens.

The *Main Information Processing Pipeline* (highlighted in Fig. 2) consists of five components:

- 1) *Perception*: Processes sensor or sensory organ data to generate an (explicit or implicit) scene representation. This includes the recognition and interpretation of road boundaries and street signs as well as the detection and behavior prediction of other road users.
- 2) *Decision Making*: Given the scene representation, this component decides which actions to take. The outputs are not motor commands, but high-level driving intents, e.g. whether to stop or not, whether to overtake, to yield, which road or lane to follow and at which speed.
- 3) *Goal and Risk Tolerance*: The decisions depend on the goals of the human or machine driver and the level of risk they are willing to take to achieve these goals.
- 4) *Execution*: The translation of the driving intent to motor commands. For an HAV, this encompasses local trajectory optimization and control.
- 5) *Body*: The physical entity of the road user, e.g. the body

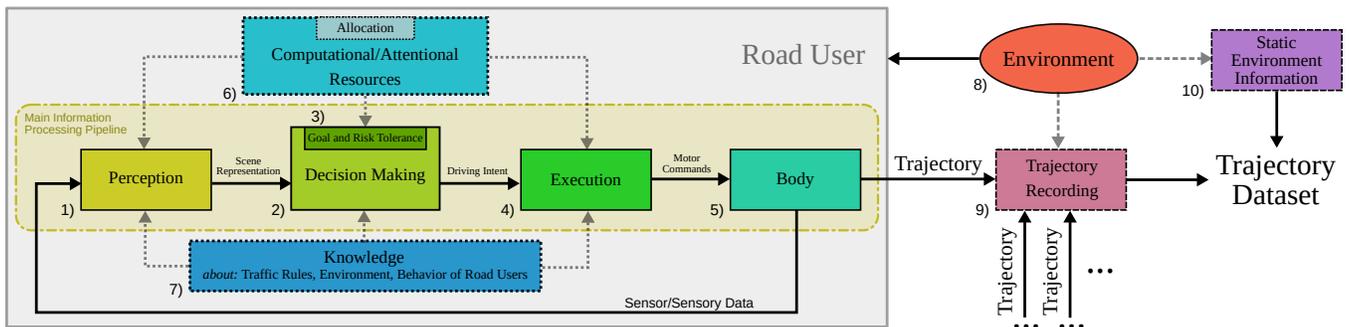


Fig. 2: Schematic of the processing pipeline for both a human road user or an automated vehicle

of a pedestrian or vehicle. The body also includes the actuation, i.e. the muscles or motors that translate motor commands to movements.

Two additional components that act on all main components complete the road user model:

- 6) *Computational/Attentional Resources*: The information processing capacity of both human brains as well as computers is limited. In psychology, these resources are termed attention resources (or simply attention) [34]. These resources can be redistributed within certain limits in both humans and computers to adapt to rapidly changing demands. In humans, this happens through a mechanism which is also just termed attention [34], [39], and in HAV through scheduling and attention mechanisms in deep neural networks.
- 7) *Knowledge*: All information processing stages require (a priori) knowledge about traffic rules, the behavior of other road users, and the physical environment (e.g. map data for localization or navigation). This knowledge can be explicitly stored in databases or long-term memory; or it can be implicit in algorithms, parameters, ML models and the datasets they are trained with, and (human) learned behaviors and experiences.

Even though more than 90% of all traffic accidents are caused by human error [40], the behavior of the road users still cannot be analyzed independently of their *Environment*. The road users sense the *Environment* through their sensors or sensory organs, but the *Environment* can also impair the observability of other road users through occlusions or adverse weather conditions. Road conditions can fundamentally change the relationship between the motor inputs and the resulting road user trajectories. Road markings, street signs, and obstructions determine which trajectories are legal and which trajectories are possible.

#### IV. TAXONOMY FOR CORNER CASES IN TRAJECTORY DATA FOR AUTOMATED DRIVING

Based on the process pipeline defined in Fig. 2, a taxonomy is developed that categorizes trajectory corner cases according to two criteria: what data is required to detect the corner case and which system component caused the corner case. Together, these two criteria span Tab. I resulting in 31 categories. The columns correspond to the four classes which describe the data required for recognition and the rows where in the system the corner case originated.

##### A. Columns

If we want to detect trajectory corner cases, we have to first consider what data is available and also what kind of data is required to classify specific kinds of corner cases we might be interested in. This might also restrict which datasets are suitable for training and has implications on the hard- and software required to detect corner cases in the wild.

Some corner cases are detectable from just the trajectory of the road user who is responsible for the corner case (subsequently termed ego). These corner cases fall into the first class *Ego Trajectory*. Corner cases of this class are typically characterized by unusual speed or acceleration values. An example of this is a car that drives at high velocity in a sharp corner, resulting in an unusually high lateral acceleration. Collisions and near-collisions with violent evasive maneuvers are also detectable by the unusual acceleration values alone and therefore also fall into this category. Inertial sensors are often sufficient to detect corner cases of this class.

Corner cases that result from the interaction of multiple road users, but do not result in a collision or an evasive maneuver, require more than one trajectory for classification. These corner cases fall into the second column of our taxonomy.

Many trajectory corner cases however can only be identified if additional environmental context is available. These corner cases fall into the third and fourth columns. Environmental context refers to any kind of information beyond the trajectory data, e.g. lane topology, obstacles, and traffic rules per lane (e.g. speed limit). Most traffic rule violations are only detectable with proper environmental context. If the ego trajectory and the environmental context are enough to detect the corner case, it is assigned to the third column. If both environmental context and the trajectories of other road users are required to detect the corner case, it is assigned to the fourth class. This is the case for violations of more complex traffic regulations like priority rules.

##### B. Rows

The rows of our taxonomy specify which system component of the ego-vehicle failed or is the cause of the unpredictable behavior, where each row in Tab. I corresponds to one of the system components described in Sec. III. This aspect of our taxonomy is similar to some previous

stage	Ego Trajectory	Ego Trajectory & Other Road Users	Ego Trajectory & Environment	Ego Trajectory & Other Road Users & Environment
1) Perception	<i>Emergency break, because deceleration of the car ahead was not registered</i>	<i>Near collision because of overlooked crossing traffic</i>	<i>Driving wrong direction in one-way street because of overlooked sign</i>	<i>Taken right of way because of overlooked sign</i>
2) Decision Making	<i>High velocity in sharp corner</i>	<i>Near collision in merge scenario due to time pressure</i>	<i>Planned disregard for a puddle that covers a large hole</i>	<i>Failure to understand precedence at a complex crossing</i>
3) Goal and Risk Tolerance	<i>Kick-down start</i>	<i>Tail gating</i>	<i>Cutting corners</i>	<i>Willingly taken right of way</i>
4) Execution	<i>Stall engine</i>	<i>Near rear-end collision because of insufficient braking</i>	<i>Scrape obstacle when parking</i>	<i>Rolling too far into an intersection due to insufficient braking while yielding</i>
5) Body	<i>Tire burst</i>	<i>Near rear-end collision due to brake failure</i>	<i>Near collision due to failing headlights</i>	<i>Taken right of way due to bad visibility of other vehicle because of broken windshield or camera optics</i>
6) Computational and Attentional Resources	Any of the examples from stages 1 – 5			
7) Knowledge	<i>High velocity in surprisingly sharp corner</i>	<i>Pedestrian at crosswalk looking away is interpreted as yielding</i>	<i>U-turn in no passing area</i>	<i>Slow approach of crossing interpreted as giving right of way</i>
8) Environment	<i>Crash into blown over tree</i>	<i>Near collision due to skidding on ice</i>	<i>Leaving the road due to Aquaplaning</i>	<i>Taken right of way due to view obstruction</i>
9) Trajectory Recording	<i>Noisy IMU measurements or failed tracking of ego road user in camera image</i>			
10) Static Environment Information	<i>Missing priority sign in dataset</i>			

TABLE I: Trajectory corner case taxonomy with example situations. The rows describe the system component in which the corner case originates. The columns identify the trajectory corner case class describing the kind of data required to detect the corner case. Ego refers to the road user that causes the corner case (multiple ego vehicles are possible if multiple trajectories are anomalous, resulting in multiple assigned categories for one corner case situation). The colors and stage number correspond to those of the components in the processing pipeline in Fig. 2.

taxonomies from traffic safety research [41], [42], [43]. These previous taxonomies are, however, strictly human driver error taxonomies as they include mechanisms or information processing stages that have no equivalent in HAVs which is comparable in terms of the behavior in case of a fault (e.g. “memory and recall” [42], “fatigue” [41], “experience” [43]). Depending on the use case, a further breakdown into subcategories can make sense. Beneficial is the distinction between *Computational/Attentional Resources* and their allocation and between different types of *Knowledge*, e.g., a priori knowledge about the behavior of other road users, the environment, and traffic rules.

It is in general not possible to unequivocally assign a single category to all situations, as multiple anomalies with distinct origins in the information processing pipelines can appear at different points of the trajectory. These anomalies often are also dependent on each other, further complicating a consistent categorization: If a driver takes another car’s right of way because they are blinded by the low winter sun, is this an *Environment* corner case due to the unfavorable lighting conditions? Or a *Perception* corner case, as a more observant driver might have been able to see the other vehicle in time? Or even a *Goal and Risk Tolerance* corner case, as a more careful driver might have approached the intersection more carefully given the environmental conditions? This is, however, not an issue specific to our corner case taxonomy. This problem was first pointed out by Senders and Moray in the context of human error taxonomies [44]. In analogy,

it also applies to human and HAV corner case taxonomies: Causal chains make selecting a single cause of the corner case context-dependent, i.e., on the application or the user’s judgment. It, therefore, can be preferable to not assign a single, but multiple categories to each corner case situation.

Another issue of attributability arises if the taxonomy is used for the classification of corner cases in trajectory datasets. The *Computational/Attentional Resources* and *Knowledge* components are frequent error sources and therefore important categories if a specific road user system is analyzed, but are not useful for the classification of corner cases in black box systems. Errors in these two components can manifest in any of the *Main Information Processing Pipeline* components. It is, therefore, often impossible to determine if the *Computational/Attentional Resources* or *Knowledge* components are the error source if only the trajectory (and environmental context) are available. Instead, for classification tasks errors in these categories have to be assigned to the components in which they manifest.

On the other hand, two additional categories become relevant for dataset classification tasks: Firstly, measurement errors like excessive noise might corrupt the trajectories during *Trajectory Recording*; secondly, mistakes could creep into the *Static Environment Information* or metadata the dataset might be enriched with. In these cases, the trajectories in the dataset could be considered corner case trajectories, although the actual ground truth trajectories might not represent corner cases at all.

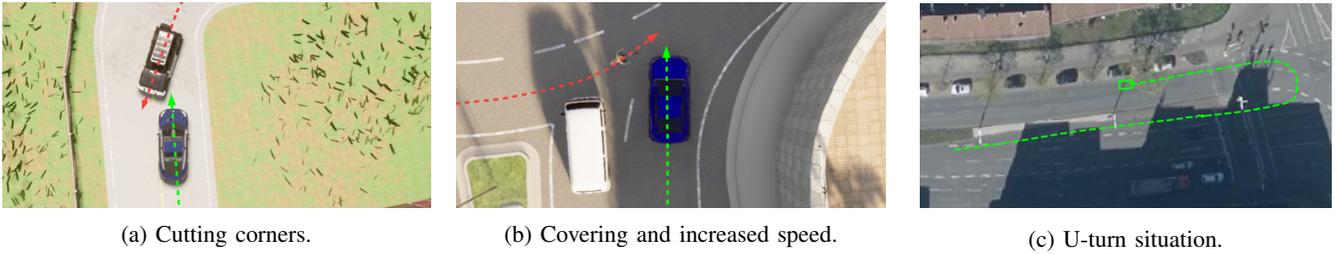


Fig. 3: Applications of the taxonomy on simulated and real world data. (a) Shows an accident event due to crossing the oncoming traffic which is classified as goal and risk tolerance with environmental reference. (b) Accident event between ego vehicle and cyclist caused by a third road user obstructing the view and the increased speed of the ego vehicle. (c) U-turn trajectory from the recordings of the KI Data Tooling project. At the last data point of the trajectory, the bounding box of the object is shown. The background image is an excerpt from the geodata of the State Office for Geoinformation and Land Surveying Lower Saxony ([www.lgln.de](http://www.lgln.de)), ©2022.

## V. APPLICATION ON DATA

After having defined the taxonomy, this chapter shows how to record and classify corner cases in real or simulation scenarios. The taxonomy is applied on several scenes with varying setups to demonstrate the generalization of the method.

### A. Application on Simulation Data

This section is about the generation of corner cases in the autonomous driving simulation software CARLA [45]. The software is an open source software for data generation and testing of ML algorithms and provides various sensor data to describe the simulated scenario, such as camera, LiDAR, and RADAR. Moreover, the software also delivers ground truth data that is cost-intensive and time-consuming to generate with real data. In addition, it offers the possibility of creating safety-critical situations in the virtual world so as not to endanger road users. In [46], a human-in-the-loop approach for generating safety-critical synthetic corner cases is presented. For this purpose, the authors have set up a test rig with two control units so that the same vehicle can be controlled by two human drivers in CARLA. Here, a semantic segmentation network is integrated into CARLA so that one of the drivers (semantic driver) only sees the networks output and thus has to control the vehicle in the CARLA world. The other driver (safety driver) monitors the situation and is supposed to intervene as soon as the semantic driver shows dangerous driving behavior. Interventions by the safety driver indicate poor situational awareness by the segmentation network and provide a safety-critical corner case. During the experiments in [46], several corner cases were recorded from which the trajectory data could subsequently be obtained. Two of them are now classified based on the categories defined in Sec. IV.

In the first scene, an accident occurs even though the ego vehicle is obeying the traffic rules due to the fact that the road user in the opposite lane is driving too fast around the curve and crosses the lane of the ego vehicle, see Fig. 3a. Due to the increased speed, the road user crosses the oncoming lane, so this situation would be classified as *Goal and Risk Tolerance* related to the environmental trajectory.

In the second scene, a vehicle is driving towards a two-lane traffic circle. Next to this vehicle is a bus, which somewhat obscures the view of the road. While the bus driver allows a cyclist to pass, the vehicle owner collides with him due to impaired visibility, see Fig. 3b. This situation is *not* classified as a *Perception* error, because the cyclist was not perceivable behind the bus. Instead, as the *Decision Making* should account for the possibility of occluded road users with priority, the error is attributed to this category.

### B. Application on Real Infrastructural Data

This section is about the detection of corner cases in real trajectory data. The investigated trajectory data were extracted from the recordings of a 14 video camera setup from the Application Platform for Intelligent Mobility (AIM) Research Intersection of the German Aerospace Center, located in Braunschweig [47]. The dataset was collected as part of the project KI Data Tooling [48] and consists of recordings with a total length of 13 hours. As for the simulation data, multiple corner cases are included in the dataset, and one of them is exemplary described in the following.

The corner case *U-turn in no-passing area* is part of the stage *Knowledge* in the class *Ego Trajectory and Environment*. Although a no-turning sign was installed in the west of the signalized research intersection, road users keep executing U-turns there. The KI Data Tooling dataset contains several U-turns, from which one of them is visualized in Fig. 3c.

The U-turn trajectory was identified by employing virtual loops as in [49]. Therefore the loops were placed manually at the entry and exit of the junction. Afterward, all intersections between the trajectories of the dataset and the virtual loops were calculated. Finally, the U-turn trajectory can be detected because, compared to other trajectories, it intersects both virtual loops.

## VI. CONCLUSION

In this article we proposed a definition of corner case trajectories in the context of automated driving and described its reference to related terms like *anomalous*, *interaction*, and *relevance*. Furthermore, we presented a model of a

processing pipeline, which is applicable to a human or automated vehicles and can be employed to categorize the causes of corner cases. Finally, we defined our taxonomy for corner case trajectories by combining the presented model with the different types of data sources that are required to detect the corner case. To demonstrate the use of the taxonomy, we gave examples of corner case trajectories in real and synthetic data.

#### ACKNOWLEDGMENT

This work results from the project KI Data Tooling (19A20001O) funded by the German Federal Ministry for Economic Affairs and Climate Action (BMWK). We gratefully acknowledge the advise of Jonas Imbsweiler.

#### REFERENCES

- [1] F. Heidecker *et al.*, “An Application-Driven Conceptualization of Corner Cases for Perception in Highly Automated Driving,” in *Proc. of IV*, Nagoya, Japan, 2021, pp. 1–8.
- [2] J. Breitenstein *et al.*, “Systematization of Corner Cases for Visual Perception in Automated Driving,” in *Proc. of IV*, Las Vegas, NV, USA, 2020, pp. 1257–1264.
- [3] S. Thrun, W. Burgard, and D. Fox, Eds., *Probabilistic Robotics*, ser. Intelligent Robotics and Autonomous Agents. Cambridge, MA: MIT Press, 2005.
- [4] M. Bieshaar, *Cooperative Intention Detection using Machine Learning—Advanced Cyclist Protection in the Context of Automated Driving*, ser. Intelligent Embedded Systems. Kassel University Press, 2021, (Dissertation, University of Kassel, Department Electrical Engineering/ Computer Science).
- [5] P. Koopman, A. Kane, and J. Black, “Credible Autonomy Safety Argumentation,” in *Proc. of SSS*, Bristol, UK, 2019.
- [6] K. Pei *et al.*, “DeepXplore: Automated Whitebox Testing of Deep Learning Systems,” *CACM*, vol. 62, no. 11, pp. 137–145, 2019.
- [7] M. Zhang *et al.*, “DeepRoad: GAN-based metamorphic testing and input validation framework for autonomous driving systems,” in *Proc. of ASE*, Montpellier, France, 2018, pp. 132–142.
- [8] J. Platon, G. Avrin, and A. Chan-Hon-Tong, “Generating corner cases for crashtesting deep networks,” in *Proc. of ECAI*, virtual, 2020.
- [9] S. Houben *et al.*, “Inspect, Understand, Overcome: A Survey of Practical Methods for AI Safety,” *arXiv preprint arXiv:2104.14235*, 2021.
- [10] J. M. Zhang *et al.*, “Machine Learning Testing: Survey, Landscapes and Horizons,” *IEEE Trans. on SE*, vol. 48, no. 1, pp. 1–36, 2022.
- [11] D. Kottke *et al.*, “Toward optimal probabilistic active learning using a Bayesian approach,” *Machine Learning*, vol. 110, no. 6, pp. 1199–1231, 2021.
- [12] J. Bolte, A. Bär, D. Lipinski, and T. Fingscheidt, “Towards Corner Case Detection for Autonomous Driving,” in *Proc. of IV*, Paris, France, 2019, pp. 438–445.
- [13] D. M. Hawkins, *Identification of Outliers*, ser. Monographs on applied probability and statistics. Dordrecht, Netherlands: Springer, 1980.
- [14] F. Gao *et al.*, “A Trajectory Evaluator by Sub-Tracks for Detecting VOT-Based Anomalous Trajectory,” *ACM Trans. on KDD*, vol. 16, no. 4, pp. 1–19, 2022.
- [15] C. Gruhl, B. Sick, and S. Tomforde, “Novelty Detection in Continuously Changing Environments,” *Future Generation Computer Systems*, vol. 114, pp. 138–154, 2021.
- [16] E. O. Eldawy and H. Mokhtar, “Clustering-Based Trajectory Outlier Detection,” *IJACSA*, vol. 11, no. 5, pp. 133–139, 2020.
- [17] F. Heidecker, M. Bieshaar, and B. Sick, “Towards Corner Case Identification in Cyclists’ Trajectories,” in *Proc. of CSCS*, Kaiserslautern, Germany, 2019, pp. 1–2.
- [18] Y. Liu, K. Zhao, G. Cong, and Z. Bao, “Online anomalous trajectory detection with deep generative sequence modeling,” in *Proc. of ICDE*, Dallas, TX, USA, 2020, pp. 949–960.
- [19] G. Markkula *et al.*, “Defining interactions: a conceptual framework for understanding interactive behaviour in human and automated road traffic,” *TIES*, vol. 21, no. 6, pp. 728–752, 2020.
- [20] J. Harris and S. R. Perkins, *Traffic conflict characteristics*. General Motors Corporation, 1968.
- [21] M. Treiber, A. Kesting, and C. Thiemann, *Traffic Flow Dynamics: Data, Models and Simulation*. Springer Berlin Heidelberg, 2012.
- [22] D. Helbing and P. Molnar, “Social force model for pedestrian dynamics,” *PRE*, vol. 51, no. 5, p. 4282, 1995.
- [23] A. Alahi *et al.*, “Social LSTM: Human Trajectory Prediction in Crowded Spaces,” in *CVPR*, Las Vegas, NV, 2016, pp. 961–971.
- [24] X. Zhang *et al.*, “Finding critical scenarios for automated driving systems: A systematic literature review,” *arXiv preprint arXiv:2202.07438*, 2021.
- [25] J. Hillenbrand, K. Kroschel, and V. Schmid, “Situation assessment algorithm for a collision prevention assistant,” in *Proc. of IV*, Las Vegas, NV, 2005, pp. 459–465.
- [26] J. Eggert, “Predictive risk estimation for intelligent ADAS functions,” in *Proc. of ITSC*, Qingdao, China, 2014, pp. 711–718.
- [27] J. Dahl, G. R. de Campos, C. Olsson, and J. Fredriksson, “Collision Avoidance: A Literature Review on Threat-Assessment Techniques,” *IEEE Trans. on IV*, vol. 4, no. 1, pp. 101–113, 2019.
- [28] D. Kamran *et al.*, “Risk-aware high-level decisions for automated driving at occluded intersections with reinforcement learning,” in *Proc. of IV*, Las Vegas, NV, 2020, pp. 1205–1212.
- [29] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA: MIT Press, 2016.
- [30] Ö. Ş. Taş *et al.*, “Making Bertha cooperate—team AnnieWAY’s entry to the 2016 Grand Cooperative Driving Challenge,” *IEEE Trans. on ITS*, vol. 19, no. 4, pp. 1262–1276, 2017.
- [31] E. Yurtsever *et al.*, “A survey of autonomous driving: Common practices and emerging technologies,” *IEEE Access*, vol. 8, pp. 58 443–58 469, 2020.
- [32] C. Berger and B. Rumpe, “Engineering autonomous driving software,” *arXiv preprint arXiv:1409.6579*, 2014.
- [33] R. Matthaei and M. Maurer, “Autonomous driving – a top-down-approach,” at *- Automatisierungstechnik*, vol. 63, no. 3, pp. 155–167, 2015.
- [34] C. D. Wickens, *Engineering Psychology and Human Performance*, J. G. Hollands, S. Banbury, and R. Parasuraman, Eds. London, UK: Routledge, Taylor & Francis Group, 2016.
- [35] D. Shinar, *Traffic Safety and Human Behavior*. Emerald Group Publishing, 2017.
- [36] M. R. Endsley, “Toward a theory of situation awareness in dynamic systems,” *Hum. Factors*, vol. 37, no. 1, pp. 32–64, 1995.
- [37] J. R. Anderson, D. Bothell, M. D. Byrne, S. Douglass, C. Lebiere, and Y. Qin, “An integrated theory of the mind,” *Psychol. Rev.*, vol. 111, no. 4, p. 1036, 2004.
- [38] D. Dörner and C. D. Güss, “PSI: a computational architecture of cognition, motivation, and emotion,” *Rev. Gen. Psychol.*, vol. 17, no. 3, pp. 297–317, 2013.
- [39] C. D. Wickens and J. S. McCarley, *Applied Attention Theory*. CRC press, 2008.
- [40] B. E. Sabej and H. Taylor, *The Known Risks We Run: The Highway*. Boston, MA: Springer US, 1980, pp. 43–70.
- [41] J. Treat *et al.*, *Tri-level Study of the Causes of Traffic Accidents: Final Report. Volume I: Causal Factor Tabulations and Assessments*. Indiana University, Bloomington, 1977.
- [42] N. A. Stanton and P. M. Salmon, “Human error taxonomies applied to driving: A generic driver error taxonomy and its implications for intelligent transport systems,” *Safety Science*, vol. 47, no. 2, pp. 227–237, 2009.
- [43] A. Khattak, N. Ahmad, B. Wali, and E. Dumbaugh, “A taxonomy of driving errors and violations: Evidence from the naturalistic driving study,” *Accident Analysis & Prevention*, vol. 151, p. 105873, 2021.
- [44] J. W. Senders and N. P. Moray, *Human Error: Cause, Prediction, and Reduction*. CRC Press, 1991.
- [45] A. Dosovitskiy *et al.*, “CARLA: an open urban driving simulator,” in *Proc. of CoRL*, Mountain View, USA, 2017, pp. 1–16.
- [46] K. Kowol, S. Bracke, and H. Gottschalk, “A-Eye: Driving with the Eyes of AI for Corner Case Generation,” 2022.
- [47] S. Knake-Langhorst and K. Gimm, “AIM Research Intersection: Instrument for traffic detection and behavior assessment for a complex urban intersection,” *JLSRF*, vol. 2, p. 65, 2016.
- [48] European Center for Information and Communication Technologies – EICT GmbH, “KI Data Tooling,” 2022, accessed 24-March-2022. [Online]. Available: <https://www.ki-datatooling.de/>
- [49] M. Junghans *et al.*, “Comparison of safety and kinematic patterns of automated vehicles turning left in interaction with oncoming manually driven vehicles,” *IEEE Trans. on TS*, 2021.