Applying Concept-Based Models for Enhanced Safety Argumentation

João Paulo Costa de Araujo*§, Balahari Vignesh Balu†§, Eik Reichmann*, Jessica Kelly†,
Stefan Kugele‡, Núria Mata†, Lars Grunske*
*Humboldt University of Berlin, Berlin, Germany
{jp.araujo, reichmae}@hu-berlin.de, grunske@informatik.hu-berlin.de
†Fraunhofer IKS, Munich, Germany
{balahari.balu, jessica.kelly, nuria.mata}@iks.fraunhofer.de
‡Technische Hochschule Ingolstadt, Ingolstadt, Germany
stefan.kugele@thi.de

***Abstract*—We consider the use of concept bottleneck models (CBMs) to enhance safety argumentation for classification tasks in safety-critical systems. When constructing a safety argumentation for Machine Learning (ML) models, there exists a semantic gap between the specified behaviour, given through class labels at training time, and the learnt behaviour, measured through performance metrics. We address this gap by using CBMs, a class of interpretable ML models in which the predictions rely on a set of human-defined concepts. A Goal Structuring Notation (GSN)-based safety assurance case is constructed including such concepts, allowing traceability between the system specification and the behaviour of the model. As a result, a line of safety argumentation is provided that relies on the interpretable model trained to satisfy the specified safety requirements.**

***Index Terms*—Concept Bottleneck Models, Semantic Gap, Safety Assurance, Interpretability**

I. Introduction

The high performance of Deep Neural Network (DNN)-based perception components makes a compelling case for their use in Automated Driving. Despite this, the task of arguing their safety prior to vehicle deployment remains a significant hurdle. Current domain-specific standards lack comprehensive guidelines for constructing safety arguments tailored to Machine Learning (ML) systems, primarily due to the challenge of specifying their inherently black-box nature.

The ISO 26262 standard [1], which focuses on functional safety, mandates that an assurance argument is made demonstrating the functional safety of the system. ISO 21448 [2] focuses instead on the safety of the intended functionality. In accordance with ISO 21448, safety argumentation is sufficient if an argument is presented showing that any output insufficiencies of the Machine Learning component will not lead to hazardous behaviour. The upcoming standard on AI Road Vehicle Safety – ISO PAS 8800 [3] provides additional guidelines on how to argue the safety of ML-components used in Autonomous Driving. Various approaches towards safety argumentation of ML components [4–9] have been developed based on Goal Structuring Notation (GSN) [10]. However, several challenges to developing such argumentations still exist. The inability to fully specify the behaviour of an ML system is one such challenge [11]. Burton et al. [12] provide insights into various gaps which might arise during the construction of a safety assurance argument, and highlight the semantic gap between specified and intended behaviour of ML functions. This gap exists due to the complexities involved in describing (i) the required specification for the ML component's functionality and (ii) the limited understanding of the black-box nature of the ML component.

In this paper, we illustrate how the use of concept bottleneck models (CBMs) [13] could address these challenges and enhance safety assurance argumentation. We show how human-understandable concepts can assist the specification of the required functionality, while gaining a deeper understanding of the inner workings of the ML model. We propose quantitative metrics based on these concepts, which aim to reduce the semantic gap and provide traceability between the different artefacts presented in a safety assurance case.

Our contributions are as follows:

- We provide a line of argumentation for a traffic sign classification component focusing on (1) the semantic gap, (2) addressing functional insufficiencies, and (3) enriched failure analysis.
- We introduce safety-relevant metrics based on human-understandable concepts which strengthen the safety argumentation.
- We demonstrate our results on a standard benchmark dataset (German Traffic Sign Dataset (GTSRB) [14]).

We exemplify our results using a safety-critical Construction Zone Recognition (CZR) sub-system of an autonomous vehicle. This sub-system operates as part of a Construction Zone Assist (CZA), which controls the vehicle when a construction zone is detected on a highway. Within the CZR, we consider the traffic sign classification ML component. When a Road-work sign is identified, the overall system prompts a hand-over of control to the driver. We propose an argumentation strategy for the traffic sign classification component. Furthermore, we

§The authors contributed equally to this paper.

This work has been financially supported by the Federal Ministry of Education and Research (BMBF) as part of MANNHEIM-AutoDevSafeOps (reference numbers 01IS22087H, 01IS22087E, 01IS22087P).

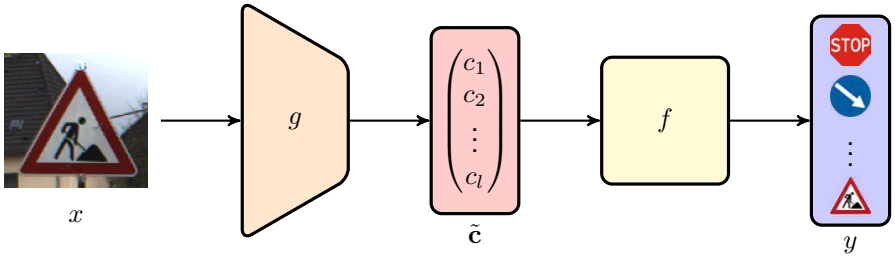

Fig. 1: Architecture of the Concept Bottleneck Model from Koh et al. [13] adapted to the task of traffic sign classification: The input x, which consists of traffic sign images, is fed into a CNN backbone g which predicts the low-level concepts in the form of a concept vector $\tilde{\mathbf{c}}$, followed by a fully connected network f which predicts the class of the traffic signs, providing the final output y.

explore the potential for concept-based models to support safety within the sub-system.

To investigate the validity of our approach and the effectiveness of concept bottleneck models for enhancing safety, our work is guided by the following three research questions (RQs):

- **RQ1:** *What is the performance trade-off of using a concept-bottleneck model when compared to a baseline CNN?*
- **RQ2:** *How do concepts impact the class label predictions, and can the interpretability of the concepts enhance safety?*
- **RQ3:** *How can the assurance argumentation benefit from the implementation of the CBM and the interpretability of the concepts?*

First, we explore the performance trade-off between a CBM and a standard convolutional neural network (CNN), and, if present, whether such a trade-off is justifiable. Then, we investigate how the learned concepts influence the model's class label predictions and how the interpretability of these concepts enhances safety by providing insights into the reasoning process. Finally, we examine how the implementation of a CBM and the interpretability of its concepts can benefit the construction of safety arguments for the ML component.

The remainder of this paper is organized as follows: Section II provides an overview of relevant background, and Section III describes related work. Section IV outlines the details of our proposed methodology for applying concept bottleneck models for enhanced safety argumentation. In Section V, we describe our experimental setup and demonstrate the results in Section VI. We discuss the results in Section VII and threats to validity in Section VIII. In Section IX, we summarize our contributions and future work.

II. Background

In this section, we first provide an overview of safety cases for ML systems, followed by an introduction to the challenges for safety argumentation, and ending with an overview of concept bottleneck models.

A. Safety Cases for ML Systems

As traditional safety-critical functions are replaced by machine learning components, providers will need to demonstrate that these systems are acceptably safe. In the automotive industry, several domain-specific standards require that a safety case is presented. For the purposes of this work, we consider the following definition of a safety case, as provided by Kelly and Weaver [15]: *"A safety case should communicate a clear, comprehensive and defensible argument that a system is acceptably safe to operate in a particular context."* ISO 26262 necessitates that a safety case is constructed demonstrating functional safety, and the upcoming ISO PAS 8800 *Road vehicles – Safety and Artificial Intelligence* requires that an assurance argument is made demonstrating that the requirements allocated to an AI-enabled function have been met with sufficient confidence. To improve the clarity and rigour of safety cases, Kelly and Weaver introduce *Goal Structuring Notation*. GSN provides a graphical representation of safety cases using goals, solutions, strategies, assumption, and context nodes, resulting in a safety case in the format of a GSN tree graph. A safety case should thus clearly describe safety requirements, arguments, and evidences, along with the relationships between them. For ML systems, it is often challenging to construct such safety cases, which we elaborate on in the following section.

B. Challenges for Safety Argumentation

One of the key challenges to safety argumentation of ML systems is the presence of the semantic gap – defined as the difference between the implicit intentions of the system's functionality and the explicit, concrete specification that is used to build the system. The semantic gap for ML systems, and corresponding implications for safety argumentation, has been discussed across literature. Burton et al. [12] identify gaps which may occur across the development process of an ML function, and identify the semantic gap. Burton and Herd [16] highlight the impact of uncertainty on the confidence in an assurance argument. They extend the taxonomy of uncertainty for an ML-system presented by Lovell [17], where the highest uncertainty stems from the specification.

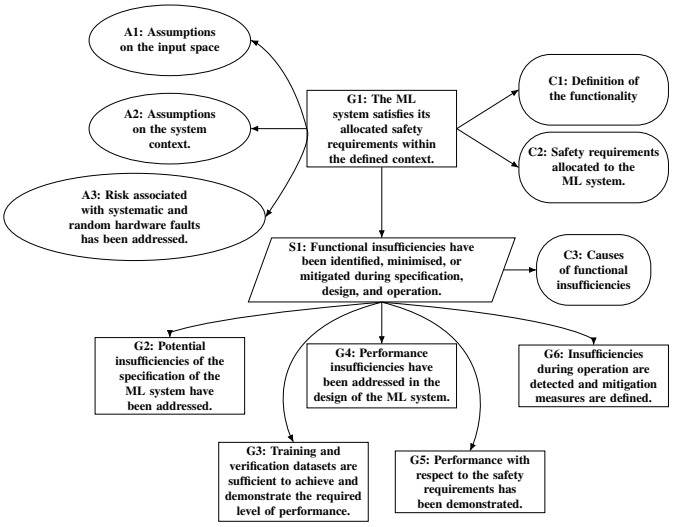

Fig. 2: As part of our safety argumentation strategy, we use the top-level safety assurance argument for a supervised ML system proposed in the work of Burton and Herd [16] as a baseline for developing the safety case.

To address the various manifestations of uncertainty in an assurance argument, Burton and Herd [16] present a structure for a generic assurance argument for a safety-relevant function implemented using supervised ML, shown in Fig. 2. The top-level goal **G1** that the ML-system satisfies its safety requirements is assured using the strategy **S1** that the insufficiencies have been mitigated during specification, design, and operation. We base our work on this top-level structure, demonstrating how concept bottleneck models can be used to address various sub-goals of **G2-G6**.

C. Iterative Approach to Safety Analysis

The development of machine learning functions requires an iterative approach to safety. The Assurance of Machine Learning for use in Autonomous Systems (AMLAS) [9] was developed with the goal of systematically integrating safety assurance cases into the ML-lifecycle. AMLAS operates in parallel with the ML development cycle, but requires predefined system safety requirements as an input. The process is structured into six stages. The first stage, called *ML Safety Assurance Scoping*, maps relevant system requirements to the ML component. Using the mapped component requirements, Stage 2, *Deriving ML Safety Requirements*, translates them into specific safety requirements tailored to the ML component. In *Defining Data Requirements* the data collection criteria are defined using the previously established ML safety requirements. Stage 4, *Model Learning*, focuses on core development activities such as training and testing. *Model Verification*, Stage 5, consists of the evaluation of the performance of the trained model on unseen data to ensure it adheres to the established safety requirements. Finally, Stage 6, called *Model Deployment*, allows for the integration of the model into the broader system. However, this step is only initiated if the model successfully navigates all aforementioned stages and demonstrates its adherence to the safety requirements. The process is iterated over and refined as new insufficiencies are discovered after deployment. The AMLAS process thus highlights the iterative approach required to construct safety arguments for ML components. In this work, we discuss how concept bottleneck models can contribute to an iterative refinement of safety assurance cases.

D. Concept Bottleneck Models

Interpretability is considered a fundamental attribute for building trust in ML systems [18]. Numerous techniques have attempted to address the black-box nature of ML functions by making their behaviour more interpretable [19]. The need for interpretability becomes even more relevant when ML functions are used in safety-critical applications [20]. Some approaches rely on using meaningful and interpretable concepts to provide a holistic view of the model's decision-making process. Koh et al. [13] define a class of models named concept bottleneck models, in which a layer is added to the architecture between the input and the final output. This bottleneck layer includes an intermediate representation of the input limited to recognising a set of predefined concepts, e. g., "shape" and "colour." The final output is then predicted based on the values of these concepts. The relationship between the concepts and the final output is modelled in a way that fosters the understanding of the decisions made by the model.

While classical neural networks directly map the input $x \in X$ to the predicted labels $y \in Y$ as $f\colon X \to Y$, a CBM first maps samples to a latent space using a concept predictor $g\colon X \to C$, where the latent space $\tilde{\mathbf{c}} \in C \subseteq \mathbb{B}^k$ represents the concept space consisting of k predefined concepts. Subsequently, a label predictor $f\colon \mathbb{B}^k \to \mathbb{R}$ maps these concepts to the final prediction. Therefore, the dataset $\mathcal{D}$ comprises data points $(x, \tilde{\mathbf{c}}, y)$, with each input x being annotated with a concept vector $\tilde{\mathbf{c}}$ and a label y. In the original paper, the authors propose three methods for training CBMs: independent, sequential, and joint training. In this work, we focus on the sequential training of CBMs, where the concept predictor g is trained independently. Afterwards, the logits of the predicted concepts $\tilde{\mathbf{c}}$ are binarized for a given threshold θ to be used in the training of the label predictor f.

III. Related Work

To the best of our knowledge, this work proposes a first application of concept-based models to enhance safety argumentation of safety-critical ML systems. Several efforts have been made to tackle the challenge of defining machine learning safety requirements which accurately encapsulate the behaviour of machine learning components. To address this problem, Rahimi et al. [21] propose a methodology for defining domain-specific concepts which better represent the behaviour of the function. The authors then propose to use post-hoc explainability methods to assess whether the ML component has learned the specified concepts. Hu et al. [22] propose an approach for specifying and testing requirements

for robustness. First, the authors derive a set of acceptable transformations on an image that are invariant or equivalent to the Human Visual System, and later assess the model performance under the recognised set of transformations. Our work makes a similar attempt to specify machine learning requirements which can be traced to model performance, and in particular traces the safety-requirements to the concepts.

The authors of Burton et al. [12] discuss that no consensus exists for addressing the semantic gap. Rather than deriving a suitable specification of the intended behaviour, several works address post-hoc explainability techniques to demonstrate that the intended behaviour was learned by the function. Methods like SHAP [23] and Saliency maps, while useful, do not fully capture the inherent decision-making processes of ML systems. To shed light on the inner workings of the model, traceability from input to output is required. Another method for post-hoc explainability which uses concepts is T-CAV by Kim et al. [24]. In T-CAV, concept activation vectors (CAV) are constructed from the latent space of an end-to-end network using input examples for different predefined concepts. Similarly, we propose to use domain-specific concepts to improve the specification of machine learning safety requirements. Rather than using post-hoc explainability methods, we leverage CBM to learn the desired behaviour implicitly.

IV. Our Approach

We propose to apply concept bottleneck models to enhance the safety argumentation and analysis of ML systems. We base our safety argumentation on the top-level GSN structure presented in the work of Burton and Herd [16]. We develop a methodology for addressing relevant sub-goals of this GSN structure by leveraging the interpretable nature of concept-bottleneck models. In the following sections, we explore how our approach contributes to the safety assurance of our CZR use-case. We focus on a traffic sign classification component of the overall system. An overview of the architecture of our concept bottleneck model for performing the task of traffic sign classification is shown in Fig. 1. Our contributions address three critical areas in the lifecycle of an ML component – (1) bridging the semantic gap between specified and intended functionality, (2) addressing functional insufficiencies at runtime, and (3) enhancing safety analysis.

A. Bridging the Semantic Gap

As discussed in Sect. II-B, one of the key challenges for safety argumentation lies in bridging the gap between specified and intended functionality, known as the semantic gap. The specified functionality describes what a system should do, and is generally outlined in the form of specification and requirement documents. A domain analysis may be conducted to gain a comprehensive understanding of the Operational Design Domain (ODD). Domain analyses support the generation of task-specific requirements, and serve as a baseline specification for ML functions. Publicly available standards such as ASAM OpenODD [25] and BSI PAS 1883 [26] provide an exhaustive list of environments, scenery information, road types, environmental factors such as weather or lighting conditions, and different objects which are used to formulate a well-defined input space for the system. The intended functionality, on the other hand, describes the behaviour of the system when it performs as it was specified to. We discuss how concept bottleneck models address the gap between specified and intended functionality through enabling an enriched specification of system behaviour.

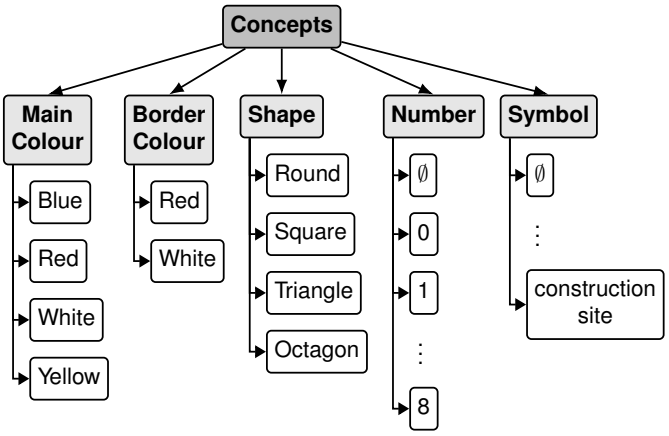

Fig. 3: The result of the input domain analysis where a set of uniquely identifiable concepts belonging to the 43 classes of the GTSRB dataset are grouped into different concept groups (highlighted in bold).

1) Specified Functionality: The specification describes the scope of the ML system. The objective of **G2** is to eliminate any insufficiencies arising from this specification. In the case of the CZR, the input space under consideration includes the set of traffic signs which need to be classified. Defining what exactly constitutes an object of a particular class is challenging, and specifications generally fail to capture these nuances. Without such a detailed specification, the model would learn the representation of different traffic sign classes from a training set. Performance criteria such as precision and recall would then be used to verify whether the learned behaviour matches the specification. Concept bottleneck models introduce the possibility to incorporate human-understandable concepts into this specification, allowing for a richer description of each class of images.

We consider our use case of traffic sign classification, and take the example of a Roadwork sign. Human understandable features of a Roadwork sign would include a white colour, a triangular shape, and a construction site symbol. Each traffic sign would have a unique set of such concepts differentiating them from one another. By analysing the set of input signs, we identified a total of 43 concepts. We categorised the sets of concepts into overarching concept groups -– main colour, border colour, shape, number, and symbol. We refer to these groups as concept groups. The complete analysis of the set of traffic signs results in a structure shown in Fig. 3. Organising the concepts into concept groups supports

in defining additional safety requirements and constraints, and provides opportunities for comprehensive safety analysis.

Using concepts allows for a more detailed description of the semantics of the input space, further enhancing the safety assurance case. We proceed with formal definitions relevant to our methodology.

Definition 1 (Concept)**.** Let $c \in \mathbb{B}$ be a variable representing an individual human-understandable or specified *Concept*. The variable c is defined as follows:

$$c = \begin{cases} 1 & \text{if the concept is associated with input image } x \\ 0 & \text{otherwise.} \end{cases}$$

Using the structure defined in Fig. 3, these individual concepts belong to a Concept Group, which is defined as follows:

Definition 2 (Concept Group)**.** Let $\mathbf{c}$ be a vector representing the *Concept Group* of size d containing individual concepts c_i, where $1 \leq i \leq d$, s.t. $\mathbf{c} = (c_1, c_2, \ldots, c_i, \ldots, c_d) \in \mathbb{B}^d$.

These concept groups are concatenated to form the Concept Vector $\tilde{\mathbf{c}}$, as follows:

Definition 3 (Concept Vector)**.** Let k be the total number of concept groups, then the *concept vector* $\tilde{\mathbf{c}}$ is defined as:

$\tilde{\mathbf{c}} = \mathbf{c_1} \oplus \cdots \oplus \mathbf{c_j} \oplus \cdots \oplus \mathbf{c_k}, \mathbf{c_j} \in \mathbb{B}^d, \tilde{\mathbf{c}} \in \mathbb{B}^l$ with $l = \sum_{i=1}^{k} |\mathbf{c_i}|$.

Using the definitions above, the dataset used for training and verification of a concept bottleneck model can be formalised in the following manner.

Definition 4 (Dataset)**.** A *dataset* $\mathcal{D}$ used for training/testing a concept bottleneck model can be defined as a set of triples

$$\mathcal{D} = \{(x, \tilde{\mathbf{c}}, y) \mid x \in X, \tilde{\mathbf{c}} \in C, y \in Y\}$$

where,

- $x \in X$ is the input image;
- $\tilde{\mathbf{c}} = (c_1, c_2, \ldots, c_l) \in \mathbb{B}^l$ is a concept vector associated with x;
- y is ground truth label associated with x;
- X is the space of input images;
- C is the space of concept vectors;
- Y is the set of ground-truth labels;

2) Intended Functionality: Goals **G4** and **G5** focus on the intended functionality. The satisfaction of these goals implies that the system is sufficient to carry out its intended task.

The objective of **G5** requires us to demonstrate that the performance of the trained model fulfils the safety requirements. Besides allowing for a better formulation of the input space, the use of concepts enables the definition of a richer set of performance criteria, enhancing Verification and Validation (V&V) activities and providing additional evidences for the safety argument structure.

We consider the following higher level safety requirement for our CZR use case:

TABLE I: An example of a high-level safety requirement formulated for the Traffic Sign Classification function.

Safety Req.	*Description*
SR1	Roadwork signs shall be detected.

For the construction zone assist use-case, false negatives are safety critical. Metrics like the recall rate are thus critical. In this case, the high-level requirement SR1 can be further refined into the requirements SR1.1 and SR1.2, shown in Tab. II.

For classical CNN models, it may be challenging to further refine this set of machine learning safety requirements. To understand the model's behaviour and assess whether it has learnt the right features, post-hoc explainability methods are often necessary to answer the question – "Has the model learnt right features?" With the use of concept bottleneck models, additional insights are possible. The previous question could be further refined to answer – "Has the model learnt the right set of concepts for a Roadwork sign?" This question could be verified through an analysis of predicted concepts at the bottleneck layer. A Roadwork sign could be uniquely identified by a set of high-level concepts such as – white background, red border, a triangular shape, and a unique construction site symbol. Considering this availability of information from the input space specification, we can expand previously established safety requirements to gain a better understanding of model behaviour.

The additional safety requirements in Table II consider the information provided by the bottleneck layer of the CBM. It is worthwhile to note that every previously defined safety requirement can also be applied for the set of relevant concept groups. Moreover, if a particular concept group is absent for a traffic sign, measuring its absence can also provide insights. In our case, the Roadwork sign would not be associated with any concept in the number group. By refining the safety goals using concept-based performance metrics, we can have a better assessment of the intended functionality. By defining safety requirements which better represent the intended functionality, we can have a more comprehensive analysis of **G5**.

The intended functionality is further impacted by the development of the ML model. The objective of **G4** is to demonstrate that the design of the chosen ML function is inherently capable of fulfilling its safety requirements. We describe how concept-bottleneck models support the satisfaction of safety requirements at design time. This goal includes any design measures and architectural considerations chosen for the ML function. The inherent black-box like nature of ML functions often leads developers to resort to explainability and post-hoc analysis methods to gain an in-depth understanding of the model and assess whether the model is capable of meeting its safety requirements. The CBM model is interpretable by nature due to its intermediate bottleneck layer. The additional insights gained from the bottleneck layer enable a deeper understanding of the model behaviour, through the definition of performance metrics which can be useful in deriving insights not only about the model, but about the dataset.

TABLE II: An example of two safety requirements obtained when using a CNN model as baseline. When using a concept based model, this list can be further extended by an additional set of 4 safety requirements.

Baseline	
Safety Req.	*Description*
SR1.1	*Recall*: $\forall x \in X$, with $y = \text{RoadWork}$, $\text{recallScore}(x) \geq \theta$ for $\theta \in [0, 1]$
SR1.2	*Precision*: $\forall x \in X$, with $y = \text{RoadWork}$, $\text{precisionScore}(x) \geq \theta$ for $\theta \in [0, 1]$
With CBM	
Safety Req.	*Description*
SR1.3	*Recall (Concepts)*: $\forall x \in X$, with $y = \text{RoadWork}$, $\text{recallScore}(c_k) \geq \theta$ for k associated with concept Main colour/White and given $\theta \in [0, 1]$
SR1.4	*Recall (Concepts)*: $\forall x \in X$, with $y = \text{RoadWork}$, $\text{recallScore}(c_k) \geq \theta$ for k associated with concept Border colour/Red and given $\theta \in [0, 1]$
SR1.5	*Recall (Concepts)*: $\forall x \in X$, with $y = \text{RoadWork}$, $\text{recallScore}(c_k) \geq \theta$ for k associated with concept Shape/Triangular and given $\theta \in [0, 1]$
SR1.6	*Recall (Concepts)*: $\forall x \in X$, with $y = \text{RoadWork}$, $\text{recallScore}(c_k) \geq \theta$ for k associated with concept $\text{Symbol/ConstructionSite}$ and given $\theta \in [0, 1]$

TABLE III: A proposed set of constraints that can be defined for effectively mitigating insufficiencies of the ML function during runtime.

Concept Group	**Rule Description**
Main colour	One concept predicted in Main colour.
Shape	One concept predicted in Shape.
Border colour	At most one concept predicted in Border colour.
Number	At most three concepts predicted in Numbers.
Symbol	At most three concepts predicted in Symbol.

B. Addressing runtime insufficiencies

After deployment, it is necessary to ensure any insufficiencies can be effectively identified and mitigated. The objective of **G6** is to ensure emerging insufficiencies during operation are adequately addressed. This goal could be addressed using plausibility checks, out-of-distribution detection, and fall-back mechanisms. In the case of traffic sign classification, the presence of a bottleneck layer allows us to enforce rules on the behaviour of the ML function. For e. g., a rule could state that if certain concepts are predicted, then the final prediction is valid, and if not a flag could be raised. The complete input space can be analysed across the identified set of concept groups in order to define rules. Some of the identified rules after analysing the input space are identified in Tab. III. The rules could be enforced as constraints on the predictions. Enforcing these constraints during testing and at runtime could assist in addressing emerging insufficiencies.

C. Enriched Failure Analysis

The safety assurance for an ML function is an iterative process. As insufficiencies are detected at runtime, new measures can be introduced to ensure the system continues to fulfill its safety requirements. When an ML function is unable to satisfy its safety requirements, the challenge becomes finding the root causes of the ML failures. The process of identifying the root cause of failures is part of the "safety analysis." Safety analyses of machine learning systems consist of iterative refinements of the system safety.

The safety analysis generally starts with analysing erroneous outputs and finding associated insufficiencies in the model. Inputs leading to the erroneous outputs are generally identified during the evaluation process. In the case of an image classification task, these inputs can be identified when the model has False Positives (FPs) and False Negatives (FNs). However, performance metrics like FPs and FNs do not give any indication of the severity of model failures, or in other words "how wrong" the ML model was. Comparing the severity of model failures caused by different inputs which were identified as FP or FN without additional insights becomes a tedious task. There is a need for safety-relevant performance metrics which provide a better indication of the severity of model failure. We introduce the hamming distance as a safety-relevant metric for concept bottleneck models. The hamming distance compares the ground truth concept vector and the predicted concept vector. The hamming distance measures how many bits have to be changed on the predicted vector to match the ground truth vector. A higher number of required changes corresponds to a higher hamming distance, indicating worse model performance. Formally, the hamming distance can be defined as follows:

Definition 5 (Hamming Distance)**.** Let $\widetilde{\mathbf{c_1}}$ and $\widetilde{\mathbf{c_2}}$ be two concept vectors in $\mathbb{B}^l$, then the *Hamming distance* h between them is defined as:

$$h(\widetilde{\mathbf{c_1}}, \widetilde{\mathbf{c_2}}) = \sum_{i=1}^{l} [\tilde{\mathbf{c}}_{\mathbf{1}i} \veebar \tilde{\mathbf{c}}_{\mathbf{2}i}],$$

where $[a \veebar b]$ denotes the bitwise XOR operation between the ith elements a_i and b_i of vectors $\mathbf{a}$ and $\mathbf{b}$, respectively.

Safety-relevant metrics can quickly pinpoint images where the model performs worse, either within each class or across entire datasets, enabling a safety-criticality-aware assessment.

While the Hamming distance can identify images which lead to safety-critical behaviour, the predicted and ground truth concept vectors offer additional insights into the model behaviour. With this information, one can assess which particular concepts are contributing towards the final prediction, which concepts are not faring well, and so on. Such insights are valuable in refining subsequent iterations of the model's safety analysis.

V. Experimental Setup

In this section, we outline our experimental setup, including details about the dataset with concept label annotations,

architectural decisions, and the training pipeline [1].

A. Implementation Details

a) Datasets: We evaluate our approach on the German Traffic Sign Recognition Benchmark (GTSRB) dataset [14]. The dataset $\mathcal{D}$ consists of a set X of images of traffic signs captured under various environmental conditions, viewpoints, lighting conditions, and with varying levels of occlusion and degradation, mirroring the complexities of real-world scenarios. Each image $x \in X$ is annotated with a ground truth label $y \in Y$, where Y is comprised of 43 traffic sign classes, with different categories, including regulatory signs (e. g., speed limits, stop signs), warning signs (e. g., pedestrian crossing, slippery road), and informative signs (e. g., direction signs, lane guidance). The images in the dataset were pre-processed by first being scaled to 224 by 224 pixels, and then standardized to have a mean of zero and a standard deviation of one. Moreover, the original split of the training and test set was kept for the experiment, with 20% of the training set being reserved as a validation set.

Definition 4 states that each image x in the CBM dataset must also be associated with a concept vector $\tilde{\mathbf{c}} = (c_1, c_2, \ldots, c_l) \in \mathbb{B}^l$, where $c_i \in \mathbb{B}$ denotes a human-understandable concept, for $1 \leq i \leq l$. We thus defined a set of concepts using an approach similar to [28, 29]. Boolean values were assigned to the traffic signs labels to indicate the presence or absence of certain high-level attributes, e. g., colours, shapes, and symbols. This process resulted in the set C of 43 concepts that were grouped into the concept groups from Fig. 3 and mapped to the input images X.

b) Architecture: As specified by Koh et al. [13], a concept bottleneck model is made up of two components: a concept predictor $g\colon X \to C$, responsible for mapping inputs $x \in X$ to their corresponding concepts vectors $\tilde{\mathbf{c}} \in C$, and a label predictor $f\colon \mathbb{B}^k \to \mathbb{R}$, which utilizes the inferred concepts $\tilde{\mathbf{c}} \in C$ to assign appropriate labels $y \in Y$. The concept predictor g leverages a pre-trained EfficientNet V2 architecture [30], with the output layer using sigmoid activation to predict the concept vectors $\tilde{\mathbf{c}}$.

The architecture of the label predictor f is structured around a feed forward neural network with two hidden layers. Each hidden layer contains 64 nodes and is activated by ReLU functions, followed by batch normalisation and dropout modules, with probability of 50%, to enhance training robustness. The final prediction $y \in Y$ is obtained through a softmax function. The batch size for both models is 128.

c) Training: As described by Koh et al. [13], the two networks were trained sequentially. Initially, the concept predictor g underwent training for 20 epochs, using the Adam optimisation algorithm [31] with an initial learning rate of 0.001, and early stopping with a patience of 5. The Binary Cross-Entropy (BCE) loss function was used, facilitating the independent assignment of a probability to each label. Following this, the label predictor f was trained for 15 epochs using the same optimisation and early stopping technique. Inputs to the label predictor are the binarised (threshold = 0.5) concept values predicted by the preceding network. Cross-Entropy loss was used with an initial learning rate of 0.01.

[1]code for reproducing results can be found in [27].

TABLE IV: A drop in performance metrics when using CBM vs a conventional CNN model as baseline.

	CBM		**Baseline**	
	Average [%]	*Top* [%]	*Average* [%]	*Top* [%]
Accuracy	97.34 ± 0.0046	98.15	99.06 ± 0.0029	99.55
Precision	96.70 ± 0.0091	97.42	98.66 ± 0.0037	99.25
Recall	95.72 ± 0.0107	97.56	98.63 ± 0.0059	99.29
F1	95.98 ± 0.0095	97.35	98.58 ± 0.0049	99.26

d) Hardware: The experiments were conducted on a system running OpenSUSE 15.5, equipped with three NVIDIA RTX 6000 GPUs, each featuring 24 GB of dedicated GDDR6 memory and a memory bandwidth of 672.0 GB/s. The GPUs operate at a core clock speed of 1770 MHz and are powered by 4608 CUDA cores, 576 Tensor cores, and 72 Raytracing cores. Additionally, the GPUs support CUDA Compute Capability 7.5, and the system is configured with CUDA, OpenCL, and TensorFlow (Python).

VI. Evaluation

In this section we present quantitative experimentation results addressing RQ1 and RQ2, offering insights into the effectiveness of our approach. The qualitative evaluation of RQ3 is left for Sect. VII.

A. Effectiveness of Concept Bottleneck Models (RQ1)

Our initial research question investigates the trade-off between performance and the interpretability inherent to CBMs. We assess this by comparing the CBM's performance against a baseline CNN model. Our goal is to explore the impact on model performance metrics when replacing a standard model with an interpretable CBM. We train a baseline Convolutional Neural Network (CNN) using the same architecture as the concept predictor of the CBM, allowing for a fair comparison between the CBM and a conventional approach to classification tasks. To ensure validity, we ran our experiment 50 times using different initialisation seeds. After each run, the metrics of accuracy, precision, recall, and F1 score were measured for each model using a held-out test set and stored for comparison. While accuracy was calculated across the entire dataset, the other metrics were assessed label-wise and averaged. Statistical significance was determined via T-tests on the metric distributions, followed by averaging and computation of standard deviations. Table IV shows the results of the experiments. Along with average results, the results for the top models are also reported. Our findings reveal a clear loss in performance when replacing a baseline model with a CBM. While the average drop in accuracy is relatively minor, the other metrics are significantly impacted when the CBM model is introduced. We discuss these results in Sect. VII.

Fig. 4: Single occurrence of a False Negative in a safety-critical sign: An example where a Roadwork sign is misclassified as a Yield sign. We notice that one of the concepts (Symbol-Construction Site) required for correctly classifying the traffic sign as Roadwork is missing.

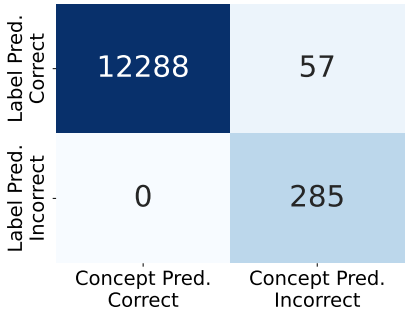

Fig. 5: Our Concept-Label alignment matrix to understand the coherence between predicted concepts from the concept bottleneck layer (CNN backbone) and final label outputs from the fully connected layers of the network. This matrix supplements our understanding of the CBM model and helps in identifying if additional fine-tuning steps are required for the first or second-half of the CBM model.

In addition to evaluating the performance metrics across all labels, we are also interested in examining the performance of the CBM for safety-critical signs. A key objective of the CZR use case is to identify images $x \in X$ in the dataset D such that the associated $y = \mathrm{RoadWork}$. Given the critical importance of avoiding false negatives in such systems, recall becomes a priority. In this scope, the average performance difference between the baseline model and the CBM is approximately 2%. When comparing the top models, the baseline achieves the highest recall score, whereas the CBM is responsible for a single false negative. Fig. 4 shows this misprediction, where the label $y = Yield$ was wrongly attributed to the image x, on the left. As mentioned in Sect. IV-C, one of the advantages of using a CBM instead of a regular CNN is the enriched safety analysis allowing for an evaluation of the model that goes beyond looking at false positives and false negatives. To showcase this advantage, a comparison between the ground truth and predicted concept vectors is displayed on the right hand side of the figure. For convenience, only the relevant concepts, i.e., concepts where $c = 1$, for $c \in \tilde{\mathbf{c}}$, were pictured. Notably, with hamming distance $h = 1$, the concept predictor g in the CBM was not able to identify the concept c_k, with k associated with $\mathrm{Symbol/ConstructionSite}$. Therefore, the label predictor f wrongly assigned the label $y = Yield$, given the match with the concepts' ground truth of the respective sign. Thus, the use of concept bottleneck models, while demonstrating a loss in overall performance, allows for a deeper analysis of misclassifications in safety-critical cases.

Summary: In addressing RQ1, we find that while the CBM model shows some performance loss compared to the baseline, fine-tuning can significantly reduce this gap, potentially down to a single misprediction in safety-critical cases. Additionally, the CBM enhances interpretability and enables deeper analysis of these scenarios.

B. Interpretability of Concept Bottleneck Models (RQ2)

Internal Coherence: To assess the interpretability of the concept bottleneck model, we first assess whether the learnt concepts correspond to the correct classes. We call this property the internal coherence of the model. Our claim is that using a CBM facilitates traceability between the components of safety arguments, as the model uses the concepts outlined in the specification to make predictions. To validate this assertion, we use the Concept-Label alignment matrix depicted in Fig. 5. This matrix offers a representation for evaluating the coherence between the predicted labels and concepts. It compares the predicted classifications of the model against the predicted concepts. The matrix is structured with rows and columns, where the row represents the correct and incorrect label predictions, while the columns denote accurate and incorrectly predicted concepts. Regarding the concepts, we consider samples as correctly predicted when all concepts in the set match the ground truth, with a single deviation resulting in the entire set being considered incorrectly predicted for that sample.

Internal coherence in a model is demonstrated when the majority of samples populate the top-left and bottom-right cells of the matrix. The top-left cell signifies instances where the model accurately predicts the label based on correct concept predictions. Conversely, the bottom-right cell highlights cases where the model incorrectly predicts the label due to inaccuracies in the predicted concepts. This emphasis on correct concept-label associations and the recognition of misclassifications due to concept errors illustrates the model's ability to maintain coherence between conceptual understanding and accurate predictions. High values in the bottom-left and top-right cells point to inconsistencies in the concept-label association. The bottom-left cell identifies samples where the labels are mispredicted despite the predicted concepts aligning with the ground truth. This poor label prediction performance may necessitate further fine-tuning of the second layer or a

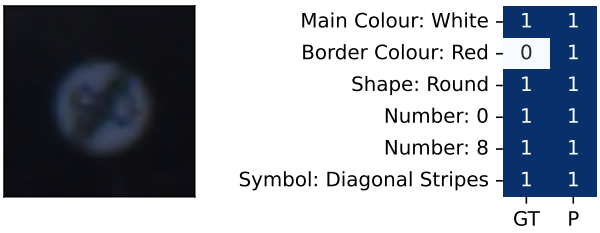

Fig. 6: Gaining additional insights from examining predicted concepts: An example where the traffic sign is correctly classified as End of Speed Limit 80 with an incorrect set of concepts. This misalignment may be attributed to the dark background, causing a border colour of red which is not present to be predicted.

Fig. 7: Gaining insights about the quality of the dataset: Using the Hamming code as an indicator of severity, the worst performing examples can be identified. The example shown here has a hamming code of 7 and upon examination, the misclassification could be attributed to the degraded quality of the image.

more precise specification of concepts. In our experiment, there were no such cases, indicating that the model correctly classifies any set of correctly predicted concepts.

Similarly, the top-right cell denotes samples where the correct label is predicted despite some mispredictions in the concept set. This cell is particularly critical, as it indicates the extent to which the model relies on concepts to make accurate predictions. High values in this cell suggest that the model's explanations may not be entirely trustworthy, potentially impacting safety. It is important to note that an incorrect prediction does not necessarily mean that all concepts were incorrect; rather, there was at least one misprediction, as previously mentioned. An example is Fig. 6, where the predicted concepts wrongly include the Border colour Red.

Enriched Failure Analysis: The use of CBMs allows us to leverage safety relevant metrics like the Hamming Distance for improved failure analysis. The Hamming Distance, presented in Definition 5, allows us to go beyond a misclassification – rather, it allows us to diagnose the severity of the model failure through capturing how many concepts the model predicted incorrectly. To demonstrate how interpretability from the bottleneck layer can be used to diagnose and debug potential safety hazards within the model, we explore two examples. In Fig. 7, we see an example where the model prediction has a Hamming Distance of 7. This means that the model was incorrect about 7 concepts. For a safety critical scenario, this metric gives us more insight into the performance than a simple misclassification. We see that the model fails to recognize any of the concepts in the image. In Fig. 8, we see an example where the model prediction has a Hamming Distance of 3. In this example, the model fails to predict all relevant concepts and thus predicts the final class incorrectly. By observing the constraints established in Tab. III, we see that in this case 2 constraints are violated. This example could thus be flagged using measures which examine constraint violations.

Summary: To address RQ2, we demonstrate that when the CBM exhibits internal coherence, safety-relevant metrics can be designed using concepts to better understand the severity of model failures.

VII. Discussion

In this section, we discuss the results for RQ1, RQ2, and RQ3.

A. *Effectiveness of Concept Bottleneck Models (RQ1)*

First, we discuss the effectiveness of replacing a classical CNN with a concept bottleneck model from a safety perspective. The significant loss in performance achieved when replacing the baseline model contradicts prevailing research in interpretable machine learning, which argues against a general trade-off between accuracy and interpretability [18, 20]. We attribute this disparity to the choice of training algorithm and to the need of further refinement of the model.

The first reason lies in the training process, particularly the use of hard values for concepts. Research by Koh et al. [13] explores different training methods, finding that using concept logits, instead of hard values, as input for the label prediction leads to optimal performance. However, this approach suffers from "information leakage" as reported by Mahinpei et al. [32] and Margeloiu et al. [33]. This leakage arises because representing binary concepts as probabilities transmits unintended information that reduces model interpretability. Current approaches mitigate this by adding hidden variables to account for the unknown information [34, 35], also at the expense of the interpretability of the model. To address this, we propose an alternative approach discussed in Sect. V that involves binarising predicted concept values. While this approach avoids the leakage, it may lead to the observed decrease in model performance.

The second reason relies on the examination of the top performing CBM, which suggests room for improvement of the model. By comparing the metrics of the top performing

Fig. 8: Identifying misclassifications using constraints: An example of a prediction that violates the set of constraints defined in Tab. III (the shape and main colour constraints are violated).

baseline model and CBM in the experiment, we notice a reduction of the discrepancy. Given that the only difference between the runs are the initialisation values, a careful selection of the model parameters could have a large impact over the final result. Additionally, we acknowledge that the current implementation of the CBM could benefit from further refinement, e. g., enhancing the preprocessing pipeline, as well as optimizing the CBM architecture, concept selection, or training strategies.

The focus of this paper is not on the performance of the CBM, but rather on arguing its benefits for enhancing safety argumentation. While there exists many methods to improve the performance of CBMs, the focus of this paper is to understand how their interpretable decision-making process can help enhance the assurance case. Therefore, we shift our discussion to the advantages provided by the interpretable nature of the CBM.

B. Interpretability of Concept Bottleneck Models (RQ2)

The evaluation of the CBM's interpretability through internal coherence analysis yielded promising results (refer to Sect. VI-B). This analysis, emphasizing the alignment between predicted concepts and true classes, supports our initial claim that CBMs facilitate the construction of traceable safety arguments. The Concept-Label alignment matrix (Fig. 5) provides a valuable tool for assessing this coherence. By scrutinizing the relationship between predicted labels and concepts, we gained insights into the model's internal reasoning. This observed alignment between concepts and classes strengthens the argument for using CBMs in safety-critical systems. Because CBMs explicitly utilize safety-relevant concepts during prediction, the reasoning steps become more transparent. This transparency allows for a more robust safety argument to be built, as each step in the reasoning process can be mapped back to the defined concepts. This traceability simplifies safety analysis and facilitates the identification of potential failure modes within the model. In Fig. 6, we show an example of a case where an incorrect set of concepts leads to a correct final prediction. These examples are particularly worrying, and indicate a lack of internal coherence in some cases. Several cases like these may indicate a need for additional safety measures.

The interpretability of CBMs also provides advantages for enhanced safety analysis. One such advantage of the CBM stems from its ability to provide insights beyond the final classification result. The evaluation of ML systems typically relies on average metrics, such as accuracy, false positive rates, or false negative rates, which may be inadequate in safety critical scenarios. To capture the performance of a model for safety-critical cases, metrics which dive deeper into the reasons for failures are necessary. Through the use of concept bottleneck models, we enable the use of safety relevant metrics which rely on concepts rather than average model performance. We presented the Hamming Distance between predicted and labelled concepts as a safety-aware metric to evaluate the performance of the model. Unlike a simple misclassification, the Hamming Distance quantifies the severity of the model's failure by measuring the number of incorrectly predicted concepts. In Fig. 8, we can see how an incorrect class prediction due to missing concept predictions may occur. While it is rational to argue that the occlusion and shadow present on the image may obstruct the identification of corresponding concepts by the CBM, the failure analysis driven by observing the Hamming code is helpful in identifying such critical examples/outliers in the datasets.

C. Contribution Towards Safety Argumentation (RQ3)

We discuss the three contributions towards safety argumentation which are enhanced by concept bottleneck models.

Specified vs Intended Functionality: The use of concept bottleneck models enables us to define a richer specification for our ML model. This enriched specification allows us to address the semantic gap between *specified and intended functionality*. We address this in two ways. First, by characterizing the dataset using human-interpretable concepts, we enrich the dataset's metadata. This additional metadata facilitates a more comprehensive analysis of data coverage. Second, by defining additional safety requirements at the concept level, we gain deeper insights into the model's behaviour, enhancing our ability to assess and improve its safety and performance.

Addressing Runtime Insufficiencies: The presence of a concept bottleneck layer introduces the possibility for better control over the model's runtime behaviour. By defining constraints on the detected concepts, we can define monitors to ensure constraints are met at runtime. This monitoring approach can support in addressing emerging insufficiencies. The ability to enforce these rules could enable the management and mitigation of potential issues as they arise, ensuring more reliable model performance in real-time scenarios. An example of identifying a misclassification through the use of constraints is shown in Fig. 8. The example violates two constraints - the shape and main colour. However, the example doesn't violate the Border colour, Symbol and Number Constraints as they were collectively defined for the entire dataset. These constraints can be adapted to suit the nature of task.

TABLE V: The added values to specific goals in the work of Burton and Herd [16] when a conventional CNN model is replaced by the CBM in the manner proposed in this paper.

Goal	Subgoal	Contribution
G2	The input domain is sufficiently well defined to ensure completeness of derived safety requirements, training and test data.	Ability to fully specify high-dimensional images through human-understandable concepts.
G4	The choice of ML technology and system design is inherently sufficient to fulfill the safety requirements.	CBM has an interpretable middle layer, enabling understanding of the model.
	Architectural measures are defined to mitigate the impact of known residual insufficiencies in the model.	Enforcing rule-based constraints in bottleneck layer.
G5	Evaluation has demonstrated that the safety-related property X is fulfilled.	Use of CBM's interpretable middle layer used to define additional safety requirements.
G6	Technical measures are sufficient to detect and mitigate residual and emerging insufficiencies during operation.	Enforcing rule-based constraints in bottleneck layer.

Enhanced Safety Argumentation: Addressing these safety aspects indirectly contributes towards different supporting arguments in the baseline GSN structure for safety argumentation introduced in Fig. 2. We select a relevant set of subgoals from goals **G2, G4, G5**, and **G6** which are enhanced by concept bottleneck models. We summarise our contributions towards these sub-goals in Tab. V.

VIII. Threats to Validity

A. Internal

A clear threat to the validity of our methodology comes from the potential for concept correlation, as identified by [29]. If the concepts used in the concept bottleneck model are highly correlated, it is challenging to discern whether the model is correctly learning individual concepts. This correlation can affect the interpretation of which specific concepts are contributing to the model's decisions. Our methodology relies on the model learning the concepts properly. Another threat to the validity of our results stems from our use of hard concepts. As identified in Sect. V, the use of hard concepts comes at the cost of lost performance. Additionally, the comparison between the CBM and a baseline CNN was performed by running several runs with different seeds. The number of runs, however, was limited by the available computational resources. Therefore, it could benefit from a few extra executions to assess the stability of the achieved results.

B. External

The main external threat to the validity of our approach is its generalizability to other use cases. Our current methodology involves a simplified traffic sign classification task where concepts are relatively straightforward to define. However, extending this approach to more complex scenarios, such as pedestrian detection, presents significant challenges. In these cases, concepts are harder to define due to the subtle nature of the features that constitute a pedestrian. To address this limitation, methodologies for systematically defining and annotating concepts in a way that can be generalized across different and more complex tasks are necessary. Without such systematic approaches, the effectiveness and applicability of concept bottleneck models may be significantly constrained in real-world, diverse applications.

IX. Conclusion

Road-vehicle standards such as ISO 26262, ISO 21448, and the upcoming ISO PAS 8800 emphasize the need for comprehensive assurance arguments which validate functional safety and intended functionality of AI/ML components. We propose to apply concept bottleneck models to enhance safety assurance argumentation. Our methodology employs human-understandable concepts for specifying functionality and gaining deeper insights into machine learning-based perception components. We apply quantitative metrics, such as the hamming distance, using these concepts to reduce the semantic gap and ensure traceability within the Safety Assurance Case. The incorporation of such safety-relevant metrics enables a comprehensive analysis of errors which goes beyond traditional metrics like the True Positive Rate (TPR) and False Positive Rate (FPR).

We argue that the enhanced interpretability offered by CBMs empowers the development of more robust safety arguments. The ability to trace reasoning steps back to specific concepts facilitates a deeper understanding of the model's decision-making process. This transparency strengthens safety arguments by allowing for a more thorough analysis of potential failure modes. Despite these efforts, the concept predictor can still be considered black-box. Emerging work on Concept Whitening [36], where the feature space is aligned with independently learned concepts, could support improvements in safety assurance for black-box ML functions.

Acknowledgment

We thank Lena Heidemann of Fraunhofer IKS for her helpful comments and discussion. We also thank the anonymous reviewers for their valuable comments and helpful suggestions. This work has been financially supported by the Federal Ministry of Education and Research (BMBF) as part of MANNHEIM-AutoDevSafeOps (reference numbers 01IS22087H, 01IS22087E, 01IS22087P).

References

[1] International Organization for Standardization, "ISO 26262:2018 Road vehicles : Functional Safety," International Organization for Standardization, Geneva, CH, Standard, Dec. 2018.

[2] ——, "ISO 21448:2022 Road vehicles : Safety of the intended functionality," International Organization for Standardization, Geneva, CH, Standard, Jun. 2022.

[3] ——, "ISO/CD PAS8800: Road vehicles : Safety and artificial intelligence," International Organization for Standardization, Geneva, CH, Standard, Jun. Under Development.

[4] S. Burton, L. Gauerhof, and C. Heinzemann, "Making the case for safety of machine learning in highly automated driving," in *SAFECOMP Workshops*, ser. Lecture Notes in Computer Science, vol. 10489. Springer, 2017, pp. 5–16.

[5] S. Burton, L. Gauerhof, B. B. Sethy, I. Habli, and R. Hawkins, "Confidence arguments for evidence of performance in machine learning for highly automated driving functions," in *SAFECOMP Workshops*, ser. Lecture Notes in Computer Science, vol. 11699. Springer, 2019, pp. 365–377.

[6] C. Picardi, C. Paterson, R. Hawkins, R. Calinescu, and I. Habli, "Assurance argument patterns and processes for machine learning in safety-related systems," in *SafeAI@AAAI*, ser. CEUR Workshop Proceedings, vol. 2560. CEUR-WS.org, 2020, pp. 23–30.

[7] E. Wozniak, C. Cârlan, E. Acar-Celik, and H. J. Putzer, "A safety case pattern for systems with machine learning components," in *SAFECOMP Workshops*, ser. Lecture Notes in Computer Science, vol. 12235. Springer, 2020, pp. 370–382.

[8] L. Gauerhof, R. Hawkins, C. Picardi, C. Paterson, Y. Hagiwara, and I. Habli, "Assuring the safety of machine learning for pedestrian detection at crossings," in *SAFECOMP*, ser. Lecture Notes in Computer Science, vol. 12234. Springer, 2020, pp. 197–212.

[9] R. Hawkins, C. Paterson, C. Picardi, Y. Jia, R. Calinescu, and I. Habli, "Guidance on the assurance of machine learning in autonomous systems (AMLAS)," *CoRR*, vol. abs/2102.01564, 2021.

[10] SCSC Assurance Case Working Group, "Goal structuring notation community standard version 3," Safety-Critical Systems Club, CA, USA, GSN Community Standard SCSC-141C, 2021.

[11] P. Koopman and M. Wagner, "Challenges in autonomous vehicle testing and validation," *SAE International Journal of Transportation Safety*, vol. 4, no. 1, pp. 15–24, 2016.

[12] S. Burton, I. Habli, T. Lawton, J. A. 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, 2020.

[13] P. W. Koh, T. Nguyen, Y. S. Tang, S. Mussmann, E. Pierson, B. Kim, and P. Liang, "Concept bottleneck models," in *ICML*, ser. Proceedings of Machine Learning Research, vol. 119. PMLR, 2020, pp. 5338–5348.

[14] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "The german traffic sign recognition benchmark: A multi-class classification competition," in *IJCNN*. IEEE, 2011, pp. 1453–1460.

[15] T. Kelly and R. Weaver, "The goal structuring notation–a safety argument notation," *Proc Dependable Syst Networks Workshop Assurance Cases*, Jan. 2004.

[16] S. Burton and B. Herd, "Addressing uncertainty in the safety assurance of machine-learning," *Frontiers Comput. Sci.*, vol. 5, 2023.

[17] B. E. Lovell, *A taxonomy of types of uncertainty*. Portland State University, 1995.

[18] C. Rudin, C. Chen, Z. Chen, H. Huang, L. Semenova, and C. Zhong, "Interpretable machine learning: Fundamental principles and 10 grand challenges," *CoRR*, vol. abs/2103.11251, 2021.

[19] P. Linardatos, V. Papastefanopoulos, and S. Kotsiantis, "Explainable AI: A review of machine learning interpretability methods," *Entropy*, vol. 23, no. 1, p. 18, 2021.

[20] C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead," *Nat. Mach. Intell.*, vol. 1, no. 5, pp. 206–215, 2019.

[21] M. Rahimi, J. L. C. Guo, S. Kokaly, and M. Chechik, "Toward requirements specification for machine-learned components," in *RE Workshops*. IEEE, 2019, pp. 241–244.

[22] B. C. Hu, R. Salay, K. Czarnecki, M. Rahimi, G. M. K. Selim, and M. Chechik, "Towards requirements specification for machine-learned perception based on human performance," in *AIRE@RE*. IEEE, 2020, pp. 48–51.

[23] S. M. Lundberg and S. Lee, "A unified approach to interpreting model predictions," in *NIPS*, 2017, pp. 4765–4774.

[24] B. Kim, M. Wattenberg, J. Gilmer, C. J. Cai, J. Wexler, F. B. Viégas, and R. Sayres, "Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (TCAV)," in *ICML*, ser. Proceedings of Machine Learning Research, vol. 80. PMLR, 2018, pp. 2673–2682.

[25] ASAM, "Asam OpenODD," Association for Standardization of Automation and Measuring Systems, 2021, accessed: 05 09, 2023.

[26] BSI, "Operational design domain (odd) taxonomy for an automated driving system (ads) - specification," British Standards Institution, 2020.

[27] J. P. Costa de Araujo, B. Balahari, E. Reichmann, J. I. Kelly, S. Kugele, N. Mata, and L. Grunske, "Repository for Applying Concept Bottleneck Models for Enhanced Safety Argumentation," Aug. 2024. [Online]. Available: https://doi.org/10.5281/zenodo.13380646

[28] J. Kronenberger and A. Haselhoff, "Dependency decomposition and a reject option for explainable models," *CoRR*, vol. abs/2012.06523, 2020.

[29] L. Heidemann, M. Monnet, and K. Roscher, "Concept correlation and its effects on concept-based models," in *WACV*. IEEE, 2023, pp. 4769–4777.

[30] M. Tan and Q. V. Le, "Efficientnetv2: Smaller models and faster training," in *ICML*, ser. Proceedings of Machine Learning Research, vol. 139. PMLR, 2021, pp. 10 096–10 106.

[31] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR (Poster)*, 2015.

[32] A. Mahinpei, J. Clark, I. Lage, F. Doshi-Velez, and W. Pan, "Promises and pitfalls of black-box concept learning models," *CoRR*, vol. abs/2106.13314, 2021.

[33] A. Margeloiu, M. Ashman, U. Bhatt, Y. Chen, M. Jamnik, and A. Weller, "Do concept bottleneck models learn as intended?" *CoRR*, vol. abs/2105.04289, 2021.

[34] E. Marconato, A. Passerini, and S. Teso, "Glancenets: Interpretable, leak-proof concept-based models," in *NeurIPS*, 2022.

[35] M. Havasi, S. Parbhoo, and F. Doshi-Velez, "Addressing leakage in concept bottleneck models," in *NeurIPS*, 2022.

[36] Z. Chen, Y. Bei, and C. Rudin, "Concept whitening for interpretable image recognition," *Nat. Mach. Intell.*, vol. 2, no. 12, pp. 772–782, 2020.