# A Unified Self-Assessment Framework for Autonomous Driving Stacks Using Subjective Logic

Thomas Wodtko*, Thomas Griebel*, Michael Buchholz, and Klaus Dietmayer

**Abstract:** Self-assessment plays a critical and important role toward safe and robust autonomous driving. Current self-assessment approaches in this area focus on individual modules at specific positions within the autonomous driving stack. The literature lacks a unifying framework to combine various self-assessment information. Hence, this work provides a comprehensive self-assessment framework for autonomous driving stacks, combining and unifying existing self-assessment methods. For this framework, we propose using subjective logic as an interface to standardize the output of self-assessment modules. This allows the combination of different modules and their use in subsequent processing modules. Our approach can be deployed to existing autonomous vehicle software stacks without imposing any requirements on their functional parts, enabling easy integration. With this framework, we are aiming to contribute to the improvement of safety and reliability in autonomous driving.

**Keywords:** Autonomous driving, self-assessment, subjective logic, subjective networks.

## 1 Introduction

In modern autonomous systems, safety and adaptability are key challenges to enable efficient yet robust processing while ensuring safety of the own system and of others. Assessment and awareness are crucial to obtain an accurate impression of the current state of a system. To move forward and bring autonomous vehicles on the road, the automotive industry has proposed functional safety standards, e.g., ISO 21448 safety of the intended functionality (SOTIF) [1]. In order to adhere to these standards, all modules within the software stack have to be assessed.

One way of assessing software modules is called self-assessment (SA). SA can be performed on different levels. Each module can be considered separately, and based on internal mechanisms and information, the current state of health is estimated. One example is the SA of filter and tracking algorithms, e.g., the Kalman filter [2, 3]. We call this *SA on module level*. In addition, by investigating the interaction of different modules, any misbehavior and issues can be identified. This can be extended to groups of modules that are assessed as a composite of modules. One example is an SA approach of multiple sensor processing pipelines as in [4]. We call this *SA on sub-system level*. Lastly, a complete system can be assessed by comparing redundant yet different approaches or

All authors are with the Institute of Measurement, Control, and Microtechnology, Ulm University, Albert-Einstein-Allee 41, 89081 Ulm, Germany {firstname}.{lastname}@uni-ulm.de

* *Thomas Wodtko and Thomas Griebel are co-first authors. Corresponding author: Thomas Wodtko.*

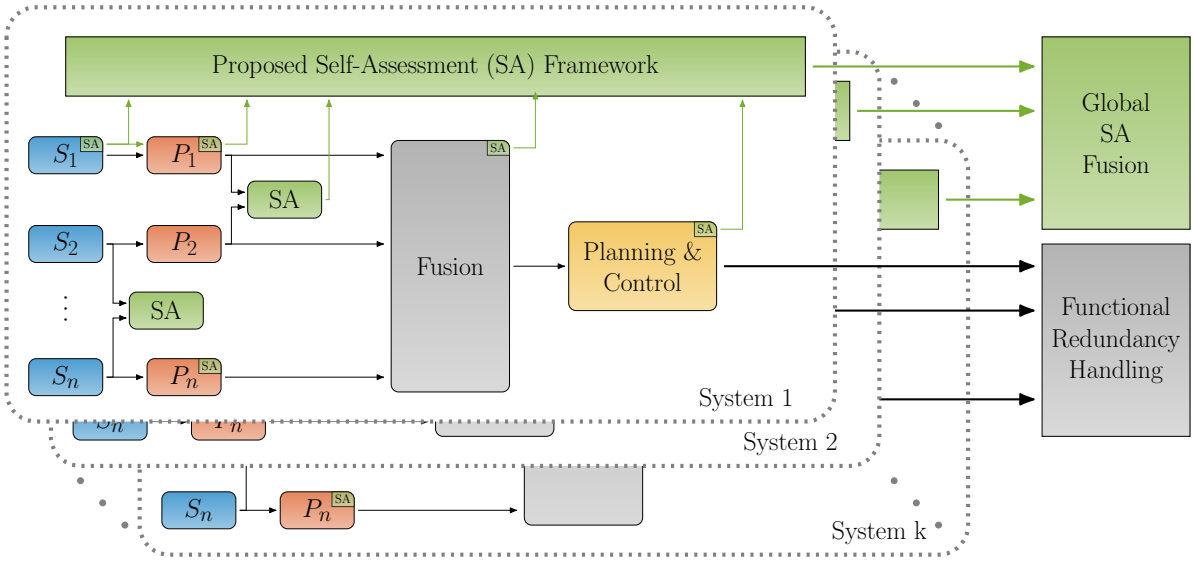



**Fig. 1.** Illustration of the overall structure of a comprehensive SA framework for a redundant system. SA is applied at different locations and levels in the autonomous driving software stack. These are *SA on module level*, here, as part of Sensor 1 ($S_1$) and Sensor Processing 1 and $n$ ($P_1$ and $P_n$), and *SA on sub-system level*, such as for the combination of $S_2$ and $S_n$ as well as $P_1$ and $P_2$. All SA results within a system are then combined in the proposed SA framework, leading to *SA on system level* representing the system-wide SA statement. The illustrated redundant system finally contains $k$ systems, and in addition to handling functional redundancy, our proposed method can be used for a global SA fusion.

pipelines that pursue the same goal as, e.g., in [5]. We call this *SA on system level*. While this is generally the most complex and computationally expensive approach, redundant systems have proven valuable for safety in autonomous systems [6].

As outlined above, various SA approaches are already available in the literature. However, these existing SA approaches primarily assess parts of the autonomous system's overall functionality individually. In addition, there is no comprehensive overview of existing SA approaches and their categorization, as we proposed above, in the literature yet. Methods and concepts that allow the combination of different outcomes from different SA levels are also missing. In this work, we propose a concept and framework that allows the combination and fusion of multiple SA approaches on all proposed levels of the autonomous driving stack, which is exemplarily illustrated in Fig. 1. With this unified SA framework, we aim to provide a foundation for future research on SA methodologies.

Therefore, we discuss existing SA approaches and categorize them in the context of the level where they are performed in the autonomous driving stack in Section 2. After introducing subjective logic (SL) theory in Section 3, we outline the similarities of these approaches, especially with respect to the SA output, and thus argue the use of SL as a common interface for SA in Section 4. The goal is to use a mathematical theory to represent and deal with SA measures in a unified manner. Hence, we present a comprehensive SA framework for autonomous driving stacks that incorporates the possible levels of SA approaches and additionally combines and fuses SA results to obtain overall statements about the whole system in Section 5. Finally, we evaluate the proposed SA framework on an exemplary system in Section 6 before we conclude our work in Section 7.

## 2 Related Work

In autonomous driving, security mechanisms are closely connected to SA approaches, which aim for safety. These methods are often called misbehavior detection systems and aim to detect misbehavior in the context of intentional attacks by an attacker. Methods in this field are discussed, e.g., in [7, 8]. However, this work focuses on the safety perspective and, thus, on SA methods.

In recent years, several ideas for single module-based SA approaches have been published (*SA on module level*). These concepts provide insights into possible assessment methods that can be applicable elsewhere. Some examples are, e.g., the SA of filtering and tracking systems by using the SL theory [2, 9, 3, 10, 11]. As outlined above, internal mechanisms are exploited and monitored to assess whether the tracking system's statistical assumptions are currently valid, such as the normalized innovation squared (NIS) consistency test [12]. Further, given the multiple different pieces of evidence on the current state of health, a framework is proposed to yield a combined SA statement for the whole tracking module, where multiple sensors can be involved [3, 10]. Another example of *SA on module level*, which in this case involves deep learning, monitors the vehicle driving performance by considering physical principles and passenger experiences [13].

Other works focused on the interaction of modules in the autonomous driving stack (*SA on sub-system level*), e.g., assessing the synchronization of sensors [14] or assessing the quality and reliability of sensor data using SL [4]. Here, the outputs of multiple modules are considered simultaneously to draw a conclusion about the collaborated result. Specifically, the method proposed in [14] obtains the synchronization state between two sensors based on their respective motion.

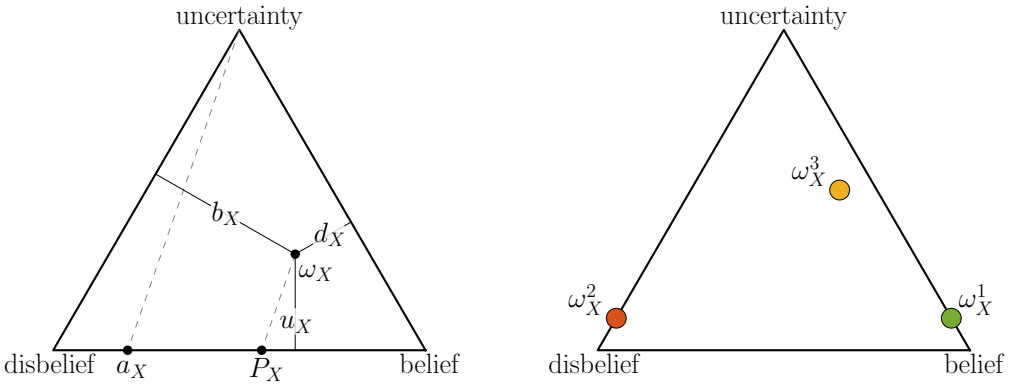
Lastly, redundancy is a well-known approach to increase safety [6] (*SA on system level*). This can even be performed between various connected and autonomous vehicles (CAVs) in a connected system using SL [5] and potentially be extended to cooperative CAVs [15]. Originating mainly from the aircraft industry [16], where sensor fault and anomaly detection have been required for years, it is also being used in cars. While the principle is simple, having redundant systems becomes complex and expensive, with more and more software and hardware being required for autonomous driving. Nevertheless, redundancy is required to meet certain safety standards [17].

It is noteworthy that several existing SA approaches over multiple SA levels in the automated driving stack use SL as the theory for building up the SA scores. Thus, we introduce SL in the following and then, with this knowledge, discuss why SL is suitable as a common interface for SA.

## 3 Subjective Logic

This section summarizes the basics of SL, which are the foundation for our proposed unified SA framework. We first present the opinion representation, the key element of SL. Second, the fusion operators relevant to this work are discussed, which combines opinions from multiple sources. Third, subjective networks (SN) are introduced, which allow the modeling of multiple sources, agents, and variables into a single representing network.

For consistency, definitions from prior studies are partially incorporated here. Each operator is cited individually, ensuring transparency and accuracy. Some definitions are




**(a)** Opinion with related measures. **(b)** Classified opinions.

**Fig. 2.** Exemplarily binomial opinions, denoted as $\omega_X$, are represented within a barycentric triangle [19]. In (a), the barycentric triangle with an opinion is specified in detail (including axes and measures). The triangle's axes correspond to belief $b_X$, disbelief $d_X$, and uncertainty $u_X$, which together define the opinion $\omega_X$. Additionally, $a_X$ represents the prior that projects $\omega_X$ onto the probability $P_X$. Moreover, exemplarily classified opinions are visualized in (b), namely a most likely true opinion $\omega_X^1$ in green, a most likely false opinion $\omega_X^2$ in red, and a somewhat true opinion $\omega_X^3$ in yellow.

taken verbatim from our previous work [18] to maintain uniform notation; however, explicit quotation marks have been omitted for improved readability. Generally, the following introduced scientific content is mainly based on [19].

## 3.1 Opinions

The key element of SL is the opinion representation. An opinion represents information about the discrete random variable $X$ from the domain $\mathbb{X}$ in terms of the belief, the uncertainty, and the a priori knowledge (base rate) about $X$.

**Definition 1 (Multinomial Opinion [19]).** *Consider a random variable $X$ in the finite domain $\mathbb{X}$ with cardinality $\mathcal{X} = |\mathbb{X}| \geqslant 2$. A multinomial opinion can be defined as an ordered triple $\omega_X = (\boldsymbol{b}_X, u_X, \boldsymbol{a}_X)$ with*

$$\boldsymbol{b}_X(x) : \mathbb{X} \mapsto [0,1], \qquad 1 = u_X + \sum_{x \in \mathbb{X}} \boldsymbol{b}_X(x)\,, \tag{1a}$$

$$\boldsymbol{a}_X(x) : \mathbb{X} \mapsto [0,1], \qquad 1 = \sum_{x \in \mathbb{X}} \boldsymbol{a}_X(x)\,. \tag{1b}$$

*The belief mass distribution $\boldsymbol{b}_X$ over $\mathbb{X}$ reflects the belief in each event, the uncertainty mass $u_X \in [0,1]$ signifies the lack of evidence, and the base rate distribution $\boldsymbol{a}_X$ over $\mathbb{X}$ represents the prior probability for each event.*

A multinomial opinion can be projected into a classical probability distribution using

$$\boldsymbol{P}_X(x) = \boldsymbol{b}_X(x) + \boldsymbol{a}_X(x) u_X\,, \quad \forall x \in \mathbb{X}\,. \tag{2}$$

Here, the projected probability $\boldsymbol{P}_X(x) : \mathbb{X} \mapsto [0,1]$ represents the expected value of the opinion when interpreted within the framework of a classical probability space.

A special case of a multinomial opinion with a binary domain $|\mathbb{X}| = |\{x, \bar{x}\}| = 2$ is called a binomial opinion. The binomial opinion $\omega_X = (b_X, d_X, u_X, a_X)$ consists of two explicit belief masses, namely the belief $b_X = \boldsymbol{b}_X(x)$ and the disbelief $d_X = \boldsymbol{b}_X(\bar{x})$. In addition, the base rate of the event $x$ is also given as a scalar: $a_X = \boldsymbol{a}_X(x)$. Note that with $a_X$, it directly follows that $\boldsymbol{a}_X(\bar{x}) = 1 - a_X$, which is, however, skipped in the opinion notation for simplicity reasons. Similarly, the projected probability of the binomial opinion $\omega_X$ is denoted as $P_X = \boldsymbol{P}_X(x)$.

Exemplarily binomial opinions using the barycentric triangle [19] are visualized in Fig. 2. In Fig. 2a, a binomial opinion with all introduced components and measures related to this opinion is depicted. In Fig. 2b, however, three specific opinions are visualized that indicate different situations. The green opinion $\omega_X^1$ visualizes a most likely true statement with a small uncertainty value $u_X$ and a large belief value $b_X$. In contrast, the red opinion $\omega_X^2$ denotes a most likely false statement with a small uncertainty value $u_X$ and a large disbelief value $d_X$. The yellow opinion $\omega_X^3$ symbols a somewhat true statement with somewhat large uncertainty and belief values $u_X$ and $b_X$.

## 3.2 Multi-Source Fusion

One of the key strengths of SL lies in its extensive framework for information fusion. SL enables the combination of multiple opinions, $\omega_X^{S_1}, \ldots, \omega_X^{S_N}$, provided by different sources $S_1, \ldots, S_N \in \mathbb{S}$, about a common random variable $X \in \mathbb{X}$. These opinions can be merged to produce a unified and comprehensive fused opinion $\omega_X^{\mathbb{S}}$.

**Definition 2 (Multi-Source Fusion [20, 21]).** *Let $\mathbb{S}$ be a set of $N \in \mathbb{N}$ sources represented by $S_1, \ldots, S_N$. Further, let $W_X^{\mathbb{S}} = \{\omega_X^{S_1}, \ldots, \omega_X^{S_N}\}$ be a set of opinions, which contains an opinion of each source about a common random variable $X \in \mathbb{X}$. Multi-source fusion describes the process of reaching a joint conclusion given the set of opinions $W_X^{\mathbb{S}}$.*

A variety of fusion operators are available in SL, as detailed in the literature [19, 20, 21]. The choice of an appropriate fusion operator depends on the specific application, the context of the situation, and the underlying assumptions [19]. In our proposed SA framework, we use cumulative belief fusion (CBF), averaging belief fusion (ABF), and weighted belief fusion (WBF), which are defined in the following. For detailed implementation steps and explicit calculation formulas for fusing multinomial opinions, please refer to [20, 21].

### 3.2.1 Cumulative Belief Fusion

The CBF [20] assumes that incorporating additional, independent sources of evidence will accumulate and strengthen the overall belief. Given $\mathbb{S}$ and $W_X^{\mathbb{S}}$ from Definition 2, the CBF of all opinions in $W_X^{\mathbb{S}}$ is denoted by

$$\oplus(W_X^{\mathbb{S}}) = \bigoplus_{S \in \mathbb{S}} \left(\omega_X^S\right) = \omega_X^{S_1} \oplus \ldots \oplus \omega_X^{S_N} \,. \tag{3}$$

Here, associativity, commutativity, and non-idempotent can be verified [20].

### 3.2.2 Averaging Belief Fusion

The ABF [20] takes into account the interdependence between sources and assumes that adding more sources does not necessarily lead to a stronger conclusion with lower uncertainty. Given $\mathbb{S}$ and $W_X^{\mathbb{S}}$ from Definition 2, the ABF of all opinions in $W_X^{\mathbb{S}}$ is denoted by

$$\underline{\oplus}\left(W_X^{\mathbb{S}}\right) = \underset{S\in\mathbb{S}}{\underline{\oplus}}\left(\omega_X^S\right) . \tag{4}$$

Here, commutativity and idempotent can be verified [20, 21]. It shall be noted that the consecutive execution of ABF of two opinions is non-associative.

### 3.2.3 Weighted Belief Fusion

The WBF [21] takes, similar to the ABF, into account the interdependence between sources and assumes that adding more sources does not necessarily lead to a stronger conclusion with lower uncertainty. However, in contrast to the ABF, the source opinions are weighted depending on the uncertainty or confidence of the opinions. Note that in the case of equally confident source opinions, the fusion is averaging. Given $\mathbb{S}$ and $W_X^{\mathbb{S}}$ from Definition 2, the WBF of all opinions in $W_X^{\mathbb{S}}$ is denoted by

$$\widehat{\oplus}\left(W_X^{\mathbb{S}}\right) = \underset{S\in\mathbb{S}}{\widehat{\oplus}}\left(\omega_X^S\right) . \tag{5}$$

As for ABF, commutativity and idempotent can be verified for WBF [21]. It shall be noted that the consecutive execution of WBF of two opinions is non-associative. To summarize, CBF assumes that integrating additional independent evidence sources will increase and solidify overall belief. Conversely, ABF considers the interdependence among sources, acknowledging that adding more sources does not necessarily lead to a stronger conclusion or reduced uncertainty. In addition, WBF is similar to ABF but fuses the source opinions depending on their uncertainty.

## 3.3 Multiplication and Co-Multiplication

In the case of binomial opinions, the multiplication and co-multiplication operators correspond to the binary logic operators AND ($\wedge$) and OR ($\vee$). These operators assume independent source opinions from two different domains, $\mathbb{X} = \{x, \bar{x}\}$ and $\mathbb{Y} = \{y, \bar{y}\}$, leading to the Cartesian product of the binary domains $\mathbb{X} \times \mathbb{Y} = \{(xy), (x\bar{y}), (\bar{x}y), (\bar{x}\bar{y})\}$. In this section, only the binomial case is considered for clarity and simplicity, as only binomial opinions are deployed in our proposed SA framework in Section 5.

### 3.3.1 Binomial Multiplication

Given two independent opinions $\omega_X = (b_X, d_X, u_X, a_X)$ and $\omega_Y = (b_Y, d_Y, u_Y, a_Y)$ on the variables $X \in \mathbb{X} = \{x, \bar{x}\}$ and $Y \in \mathbb{Y} = \{y, \bar{y}\}$, respectively, then the binomial multiplication [19] (conjunction $x \wedge y = \{(xy)\}$) is denoted by $\omega_{X\wedge Y} = \omega_X \cdot \omega_Y$.

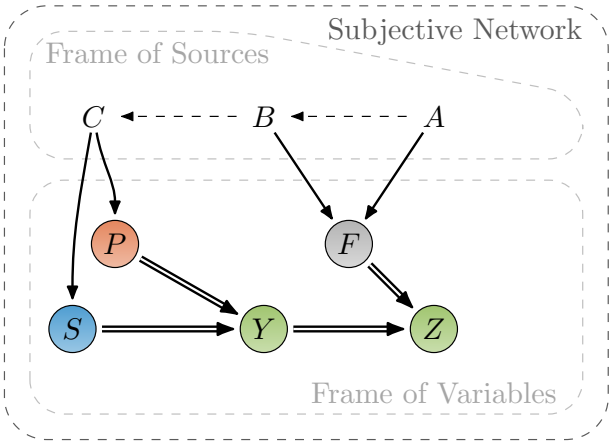



**Fig. 3.** Illustration of an exemplary SN. In this example, agent $A$ aims to decide about the variable $Z$ given the other agents $B$ and $C$, the observer $S$, $P$, and $F$, as well as the other variable $Y$. Arrows visualize the connections between these components. Namely, dashed arrows ($\dashrightarrow$) indicate trust relations, solid arrows ($\rightarrow$) represent belief relations, and double arrows ($\Rightarrow$) denote conditional relations.

### 3.3.2 Binomial Co-Multiplication

Given two independent opinions $\omega_X = (b_X, d_X, u_X, a_X)$ and $\omega_Y = (b_Y, d_Y, u_Y, a_Y)$ on the variables $X \in \mathbb{X} = \{x, \bar{x}\}$ and $Y \in \mathbb{Y} = \{y, \bar{y}\}$, respectively, then the binomial co-multiplication [19] (disjunction $x \vee y = \{(xy), (x\bar{y}), (\bar{x}y)\}$) is denoted by $\omega_{X \vee Y} = \omega_X \sqcup \omega_Y$.

## 3.4 Subjective Networks

In [19], SNs are presented as a graph-based framework that connects agents or sources with variables alongside conditional and trust opinions.

**Definition 3 (Subjective Network [19]).** *An SN describes a decision process and is a directed acyclic graph (DAG) that contains agents $\mathbb{S}$ and variables $\mathbb{V}$. The connections between agents represent trust, between agents and variables observations, and between variables conditional connections.*

An overview of the SN concept is shown in Fig. 3. To sort the components, the agents $A, B, C \in \mathbb{S}$ are in the so-called *frame of sources*, whereas the variables $S, P, F, Y, Z \in \mathbb{V}$ are in the *frame of variables*. Here, a simple SN is visualized, where agent $A$ aims to decide on the variable $Z$. For the decision process, agent $A$ has additional trust relations to agents $B$ and thus $C$. These agents have belief relations to the observed variables $S$, $P$, and $F$. Then, $S$, $P$, and $F$ have conditional relations to the variables $Y$ and $Z$.

For the implementation of the SN in Fig. 3 and, thus, the calculation of the decision process about variable $Z$, SL opinions and operations are needed. Various SL operations, presented above, can be applied to nodes of variables in this SN with conditional relations, such as $S$ and $P$ connected with $Y$, to fuse and combine the information involved. However, to choose the most suitable SL operation, the details of the situation, the considered system, and the dependencies need to be taken into account. These implementation choices are extensively discussed and explained in Section 5 while modeling and implementing our SA framework for the considered system. For more details about SN, please refer to [19].

For completeness reasons, we briefly introduce the deduction operator [19]. In some cases, deduction is required to implement conditional relationships between random variables in SNs, e.g., $Y$ and $Z$ in Fig. 3. Consider $\omega_Y$ being an opinion containing the information obtained about the random variable $Y$. Now, let $\boldsymbol{\omega}_{Z|Y} = \{\omega_{Z|y_i} \,|\, i = 1, \ldots, \mathcal{Y}\}$ be the conditional opinions of the random variable $Z$ for each possible event in $\mathbb{Y}$ Then $\omega_{Z||Y}$ represents the deducted option on $Z$ given $Y$, denoted by

$$\omega_{Z||Y} = \boldsymbol{\omega}_{Z|Y} \circledcirc \omega_Y \,. \tag{6}$$

# 4 Common Interface for Self-Assessment

As discussed in Section 2, several existing SA approaches already leverage SL, producing SL opinions as their output. This demonstrates the suitability of SL as a foundational framework for SA due to its ability to express degrees of belief, disbelief, and uncertainty in a mathematically rigorous manner. In particular, the approach introduced in [9] highlights the practical use of SL opinions, incorporating reference opinions for evaluating SA statements. Additionally, a threshold derivation method is presented, enabling informed SA decisions based on the derived SL opinions. This strengthens the case for adopting SL in SA frameworks, as it facilitates decision-making through well-defined mathematical thresholds.

Beyond its theoretical strengths, SL offers practical advantages: It provides a concise yet expressive representation of SA statements, is intuitive, and easy to process computationally. This makes SL particularly useful in real-world applications, where efficient assessment of system states is crucial. Furthermore, integrating SN extends the applicability of SL in SA. For example, our automated driving stack (illustrated in Fig. 1) can be directly modeled as an SN (see Section 5), providing a structured mathematical representation of the system. This enables the system to evaluate its own reliability dynamically, considering dependencies between different components. Lastly, SNs' flexibility allows for the modeling of various scenarios and patterns within an SA framework. By combining SL and SNs, we propose a robust and scalable approach to self-assessment, ensuring both formal rigor and practical applicability in complex decision-making systems. Thus, we propose using SL in connection with SN as a common interface for SA methods in an automated driving stack.

# 5 Self-Assessment Framework

Following the previous argumentation, we propose a unified SL-based SA framework in this section. This framework is introduced using a lucid system structure for simplicity and illustration reasons. Omitting redundancy allows for a more intuitive presentation of the framework. However, as discussed later, the applicability is not limited. The lucid system used for introducing our SA framework comprises three sensors with respective processing concatenated with a fusion and a planning & control module. It is illustrated in Fig. 4 and will also be used for our subsequent evaluation in Section 6.

In the following, a mathematical representation of the lucid system is derived using SL, especially SNs. Recurring patterns are elaborated based on the obtained SN's structure, aiming to provide fundamental components that can be combined to represent any system.

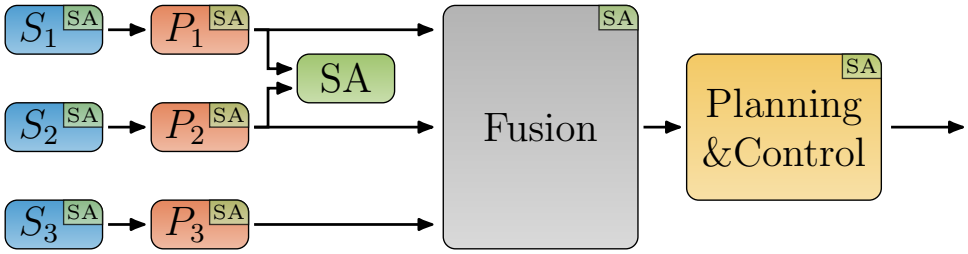



**Fig. 4.** Lucid system with a simplified structure. Three sensors ($S_1$, $S_2$, and $S_3$) with respective processing modules ($P_1$, $P_2$, and $P_3$) are present. Each module is assessed separately, and additionally, the combined sensor processing output of $P_1$ and $P_2$ are assessed using a concurrent SA module. All data is fused in the fusion module before being used for planning & control.

At several points in the presented SN, information is fused. For this, task-specific fusion operators for each pattern are discussed. Finally, the derived SN, in combination with suitable fusion operators, represents the decision process of the unified SA framework.

It should be mentioned that for simplicity reasons, the use of trust is facilitated in this work. Although explicitly shown in SNs, trust is always assumed to be dogmatic consent. Trust relations in SL enable sophisticated conflict handling [20, 18], which can potentially improve fusion within the SA framework. However, because introducing an initial approach to deploying SA on a system-wide level focuses on intuition and clarity, these advanced conflict-handling fusion techniques exceed the scope of this work. Further, in this work, we assume that the provided SA modules already take time-dependent information into account, and thus, our unified SA framework may not filter any intermediate state over time.

## 5.1 Mathematical Representation

To be represented in an SN, a system with its SA framework structure must first be analyzed in more detail. The only source of information considering SA is the SA output from each individual module or concurrent SA modules. Hence, each such output is considered as a variable $X$ inside the SN represented by an SL opinion $\omega_X$. Any other intermediate SA observation is a combination or fusion of these opinions. As previously described, opinions must describe the same random variable in order to yield interpretable fusion results. However, at first, each SA output only describes the state of the module itself (*SA on module level*) or the combined state of some group of individual modules, but not directly the overall system state (*SA on sub-system level*). Respectively, to fuse available information, virtual variables and conditional dependencies from other variables must be defined.

For example, given a sensor $S$ and its processing $P$ together with their module-specific SA opinions $\omega_S$ and $\omega_P$, respectively, the virtual variable $V$ with its opinion $\omega_V$ describes the combined state of health. Domains must be considered to obtain information about $V$. At first, each SA score only describes a specific module. Either the chosen operator must support opinions of different domains, such as multiplication, or the statistical connection must be explicitly considered. For the later, the conditionals $\boldsymbol{\omega}_{V|S}$ and $\boldsymbol{\omega}_{V|P}$ describe the effect of $S$ and $P$ onto $V$, as described with Eq. (6). For completeness reasons, these

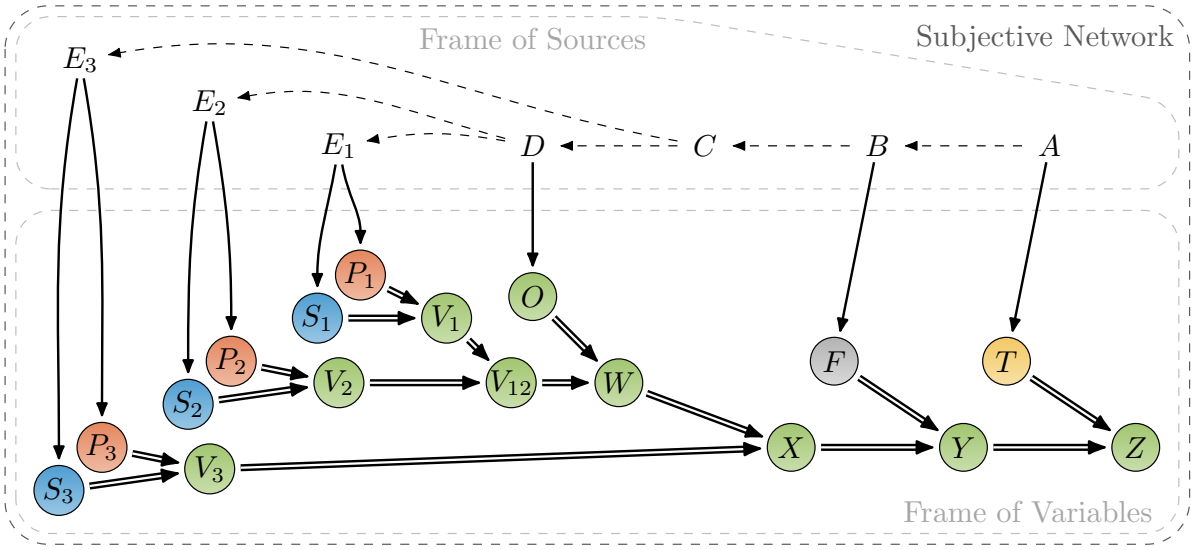


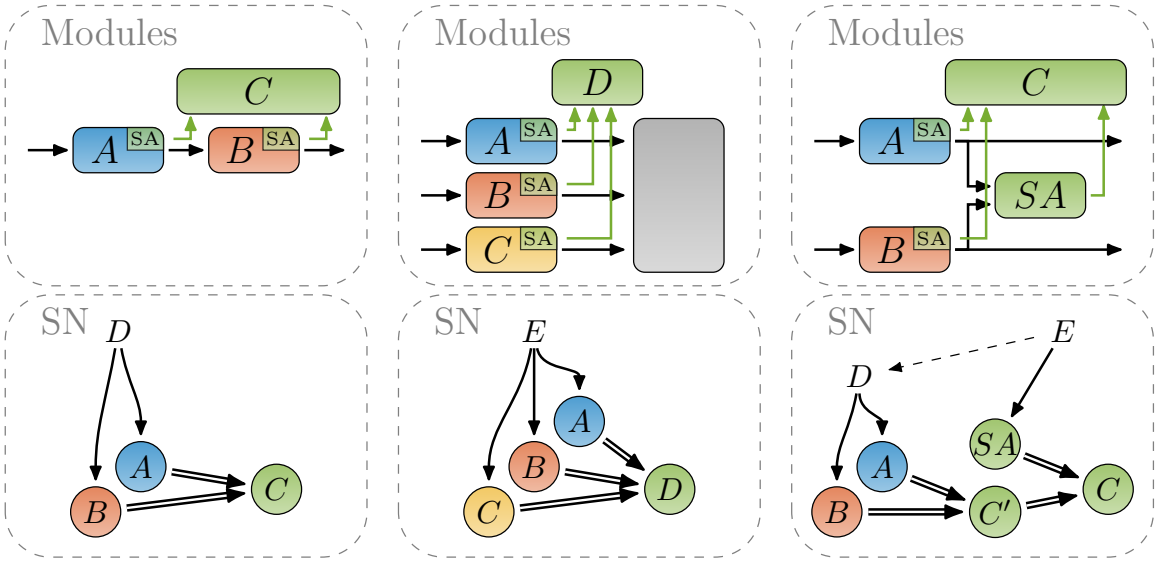

**Fig. 5.** An SN representing the lucid system from Fig. 4. Here, agent $A$ decides on the overall SA described by the variable $Z$. Virtual variables ($V_{\{1,2,3,12\}}$, $W$, $X$, $Y$, and $Z$) and virtual agents ($B$, $C$, $D$, and $E_{\{1,2,3\}}$) were added to provide a graphical representation similar to the previous system illustration. The variables $S_{\{1,2,3\}}$, $P_{\{1,2,3\}}$, $F$, and $T$ represent the SA opinions of the sensor, processing, fusion, and planning & control modules, respectively. Further, the concurrent SA opinion is represented by $O$.

conditionals are explicitly denoted and carried out within this section. However, as it will be discussed later, under some restrictions, some of these conditionals are assumed to represent identities and, therefore, will vanish in the final expression. Further, virtual agents $V$ are added to the SN to preserve an intuitive structure. Partially, they directly observe variables, or otherwise, they combine available information without any direct observation. The trust between virtual agents is dogmatic true; no data is changed by adding them, which is important when they are added for structural reasons only. In Fig. 5, the exemplary system from Fig. 4 is completely translated into a mathematical representation as an SN. The frame of sources $\mathbb{S}$ contains the virtual agents $B, C, D, E_{\{1,2,3\}}$ and the observing agents $A$. Further, the frame of variables $\mathbb{V}$ comprises the virtual variables $V_{\{1,2,3,12\}}, W, X, Y$, the observed variables $S_{\{1,2,3\}}, P_{\{1,2,3\}}, F, T$, and the variable $Z$, which $A$ decides on. In the next section, when specific patterns are defined, the availability and suitability of operators are discussed, considering different possible system configurations.

## 5.2 Building Patterns

Considering the derived SN in the previous section, certain patterns and structures appear repeatedly. In this section, these patterns are elaborated on, and possible fusion operators are evaluated and discussed depending on the specific situation in which a pattern appears. A general overview of the patterns is given in Fig. 6. Here, the structure of modules in the system overview (upper row) is shown alongside the respective SN (lower row).

While the representation of certain structures is similar at different positions in the software stack, the way the information is fused strongly depends on the specific information. In particular, choosing a fitting fusion operator is key and must consider whether, for




**(a)** Series connection. **(b)** Parallel connection. **(c)** Concurrent self-assessment.

**Fig. 6.** Different patterns of module structures in the system overview with their corresponding mathematical representation as subjective networks (SNs) are illustrated. Each part shows basic structures that can be used to assemble an implementation of any system.

example, data is statistically independent. Thus, in the following, the presented patterns from Fig. 6 are analyzed in detail, and fusion operator feasibility is discussed.

### 5.2.1 Series Connection

This section explains the series connection pattern in Fig. 6a. Two modules connected in series are, for example, a sensor acquisition and a respective processing. Usually, the SA output of the combination depends on the well-being of both modules separately. In other words, both modules must perform as expected to yield a healthy SA output. If any of both SAs report an issue, the combined output most likely shows reduced quality. Thus, the SA output must be connected using a logical AND. For this purpose, extending the AND operation, SL provides a multiplication operator [19], introduced in Section 3.3.1. If, in contrast, a module connected in series can compensate for its preceding module, the connection can be interpreted as being parallel with respect to SA and, thus, refers to Section 5.2.2.

Considering the series connection in Fig. 6a and given the two SA opinions $\omega_A^D$ and $\omega_B^D$ observed by agent $D$, the combined SA information $\omega_C^D$ when module $A$ and $B$ are connected in series is given by

$$\omega_C^D = \omega_A^D \cdot \omega_B^D \,. \tag{7}$$

The multiplication operator takes different domains into account, such as the different domains $A$ and $B$, so they do not need to be considered separately.

### 5.2.2 Parallel Connection

With multiple modules in parallel, as illustrated in Fig. 6b, the feasibility of fusion operators depends on the situation. The type of fusion operation selected for parallel connection

affects the meaning of the overall SA output. If the success requires all parallel modules to be healthy, the multiplication operator has to be applied. This implies that every part of the parallel connection is healthy individually. However, in many applications, parallel structures present a certain degree of redundancy or compensation. Respectively, their SA information individually indicates if their combination is of sufficient quality. If a single healthy module is sufficient, the co-multiplication operator, introduced in Section 3.3.2, is appropriate. Equally to the multiplication operator, it implicitly takes the different domains into account. Figuratively speaking, using co-multiplication, the combined SA statement describes minimal feasibility. As long as at least one part is healthy, the combined parallel connection is considered healthy.

Considering the parallel connection in Figure 6b and given the three SA opinions $\omega_A^E$, $\omega_B^E$, and $\omega_C^E$ observed by agent $E$, the combined SA information $\omega_D^E$ using co-multiplication is calculated by

$$\omega_D^E = \omega_A^E \sqcup \omega_B^E \sqcup \omega_C^E \,. \tag{8}$$

In contrast, if the different modules can compensate for impairments of other modules and the overall SA output should reflect issues of individual modules, fusion operators presented below should be used. First, if data is independent and each SA information provides more evidence, the CBF operator from Section 3.2.1 is appropriate. Given the SA opinions $\omega_A^E$, $\omega_B^E$, and $\omega_C^E$ observed by a agent $E$, the combined SA opinion $\omega_D^E$ using the CBF is calculated by

$$\omega_D^E = \left(\boldsymbol{\omega}_{D|A} \circledcirc \omega_A^E\right) \oplus \left(\boldsymbol{\omega}_{D|B} \circledcirc \omega_B^E\right) \oplus \left(\boldsymbol{\omega}_{D|C} \circledcirc \omega_C^E\right) . \tag{9}$$

Here, the order is irrelevant since the CBF is commutative and associative. Thus, multiple concatenated CBF operations can be reordered or combined.

If, however, individual SA information is not independent, the ABF from Section 3.2.2 suits best. Given $\omega_A^E$, $\omega_B^E$, and $\omega_C^E$ as before, the combined SA opinion $\omega_D^E$ using ABF is calculated by

$$\omega_D^E = \left(\boldsymbol{\omega}_{D|A} \circledcirc \omega_A^E\right) \underline{\oplus} \left(\boldsymbol{\omega}_{D|B} \circledcirc \omega_B^E\right) \underline{\oplus} \left(\boldsymbol{\omega}_{D|C} \circledcirc \omega_C^E\right) . \tag{10}$$

Although the ABF is commutative, it is not associative. Thus, multiple ABF operations cannot be combined.

Another suited fusion operator is the WBF from Section 3.2.3 that, similar to the ABF, considers dependent SA information. In contrast to the ABF, the WBF has a neutral element, the vacuous opinion $\omega_X$ with $u_X = 1$, and thus, when an opinion is fused with the vacuous opinion, it remains unchanged. As a result, ABF is better suited when all paths fused in the proposed framework are mandatory; if any statement of any SA module is vacuous, the resulting uncertainty becomes higher. On the other hand, if paths are optional, and a path without a meaningful SA statement is tolerable, the WBF can reflect this. Given $\omega_A^E$, $\omega_B^E$, and $\omega_C^E$ as before, the combined SA opinion $\omega_D^E$ using WBF is calculated by

$$\omega_D^E = \left(\boldsymbol{\omega}_{D|A} \circledcirc \omega_A^E\right) \widehat{\oplus} \left(\boldsymbol{\omega}_{D|B} \circledcirc \omega_B^E\right) \widehat{\oplus} \left(\boldsymbol{\omega}_{D|C} \circledcirc \omega_C^E\right) . \tag{11}$$

### 5.2.3 Concurrent Self-Assessment

In the case of concurrent SA modules, as illustrated in Fig. 6c, information is partly dependent. Thus, it presents a special case as a combination of the above and is therefore considered its own pattern. Generally, the SA of modules $A$ and $B$ can be fused similarly to the parallel case using, for example, CBF or (co-)multiplication. Although the concurrent SA is also in parallel, it is not independent, and thus, the choice of fusion operators is limited. As a result, these two fusion operations may not be combined. Depending on whether both $A$ and $B$ have their own SA, the concurrent SA may be considered optional or mandatory. Respectively, a suitable operator is presented in the previous section.

Given two modules $A$ and $B$ with their SA opinions $\omega_A^D$ and $\omega_B^D$ observed by agent $D$ and an optional concurrent SA module with output $\omega_{SA}^E$, the combined SA opinion $\omega_C^E$ observed by agent $E$ using co-multiplication and WBF is calculated by

$$\omega_C^E = \left(\boldsymbol{\omega}_{C|C'} \circledcirc \left(\omega_A^D \sqcup \omega_B^D\right)\right) \mathbin{\widehat{\oplus}} \left(\boldsymbol{\omega}_{C|SA} \circledcirc \omega_{SA}^E\right) . \tag{12}$$

Here, introducing the agent $D$ and variable $C'$ allows for a cleaner representation and separation of individual fusion steps.

### 5.2.4 Operator Selection

To implement any of the above connection patterns, a fusion operator must be chosen that correctly reflects the required constellation. In the previous section, we presented possible operators and described certain aspects that need to be considered when integrating them into the proposed unified SA framework. There is not just a single rule telling which operator to choose; it remains a process of choosing the best-fitting one for each application. While the list of available operators is not complete, we are convinced that the operators presented above can describe a wide variety of systems.

## 5.3 Assembling a System-Wide Self-Assessment Formula

In this section, the SN (cf. Fig. 5) of our lucid system (cf. Fig. 4) is combined with the fusion operators for different patterns (cf. Fig. 6) discussed in the previous section. The goal is to assemble two formulas that reflect the *overall state of health* and a *safety-critical check* of the system, respectively, only based on the patterns presented in Fig. 6. Assuming independence, conditionals are presumed to be identities and thus do not influence any opinion's belief distribution. For simplicity reasons, conditionals are therefore omitted in the following. Hence, whenever two opinions about different random variables are fused, an identity conditional is implicitly assumed.

The *overall state of health* should reflect two key aspects. One aspect is if the system provides a critical path of mandatory modules. Another one is the safety margin, which refers to the available redundancy at parallel points in the software architecture. As an example of the latter, if a sensor fails in our lucid system, the *overall state of health* should output a reduced score, even if enough information is available for safe operation.

In contrast, the *safety-critical check* only reflects the first aspect. One of all optional modules at a parallel point in the software stack is sufficient. With respect to the lucid system, a high SA score should be obtained even if two of the three sensors fail. Then, if the third one reports issues, the *safety-critical check* SA score should abruptly decrease.

Next, starting from the perception part of the SN, the two formulas are derived. The SA of the sensor and processing combinations ($S_1, S_2, S_3$ and $P_1, P_2, P_3$) can be fused using the series connection pattern (see Section 5.2.1). Respectively, the observations of the virtual agents $E_1, E_2, E_3$ are defined by

$$\omega_{V_i}^{E_i} = \omega_{S_i}^{E_i} \cdot \omega_{P_i}^{E_i}\,, \quad \forall i \in \{1,2,3\}\,. \tag{13}$$

Then, applying the concurrent SA connection (see Section 5.2.3), the SA of sensor and processing $V_1$ and $V_2$ are combined with the concurrent SA. Since an SA score is available for each module independently, the concurrent SA module is considered optional. Thus, the WBF operator is reasonable for fusion. The sensor and processing combinations $(V_1, V_2)$ are combined using CBF for the *overall state of health* and co-multiplication for the *safety-critical check*. Respectively, the combination is defined by

$$\omega_W^D = \begin{cases} \left(\omega_{V_1}^{E_1} \oplus \omega_{V_2}^{E_2}\right) \mathbin{\widehat{\oplus}} \omega_{SA}^D\,, & \text{for state of health,} \quad (14a) \\ \left(\omega_{V_1}^{E_1} \sqcup \omega_{V_2}^{E_2}\right) \mathbin{\widehat{\oplus}} \omega_{SA}^D\,, & \text{for safety-critical.} \quad (14b) \end{cases}$$

The third sensor and processing combination $(S_3, P_3)$ is fused with the other two using CBF or co-multiplication for the two formulas, similar to before. Hence, the perception SA $\omega_X^C$, consisting of all sensors, their processing, and the concurrent SA, is defined by

$$\omega_X^C = \begin{cases} \left(\left(\omega_{V_1}^{E_1} \oplus \omega_{V_2}^{E_2}\right) \mathbin{\widehat{\oplus}} \omega_{SA}^D\right) \oplus \left(\omega_{S_3}^{E_3} \cdot \omega_{P_3}^{E_3}\right)\,, & \text{for state of health,} \quad (15a) \\ \left(\left(\omega_{V_1}^{E_1} \sqcup \omega_{V_2}^{E_2}\right) \mathbin{\widehat{\oplus}} \omega_{SA}^D\right) \sqcup \left(\omega_{S_3}^{E_3} \cdot \omega_{P_3}^{E_3}\right)\,, & \text{for safety-critical.} \quad (15b) \end{cases}$$

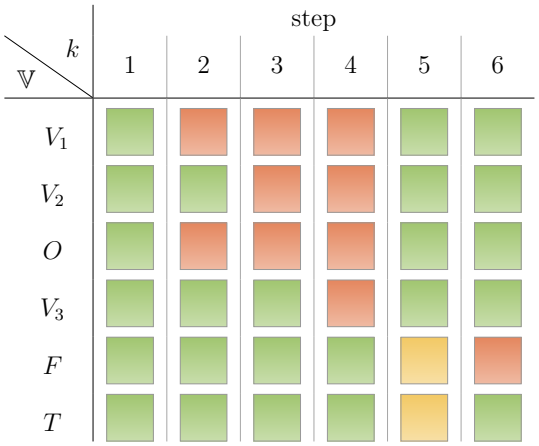
Finally, the perception is combined with fusion and planning & control. Since all three parts are mandatory and in series connection, the multiplication operator is used. The complete decision formula for the *overall state of health* and the *safety-critical check* is given by

$$\omega_Z^A = \begin{cases} \left(\left(\left(\omega_{V_1}^{E_1} \oplus \omega_{V_2}^{E_2}\right) \mathbin{\widehat{\oplus}} \omega_{SA}^D\right) \oplus \left(\omega_{S_3}^{E_3} \cdot \omega_{P_3}^{E_3}\right)\right) \cdot \omega_F^B \cdot \omega_T^A\,, & \text{for state of health,} \quad (16a) \\ \left(\left(\left(\omega_{V_1}^{E_1} \sqcup \omega_{V_2}^{E_2}\right) \mathbin{\widehat{\oplus}} \omega_{SA}^D\right) \sqcup \left(\omega_{S_3}^{E_3} \cdot \omega_{P_3}^{E_3}\right)\right) \cdot \omega_F^B \cdot \omega_T^A\,, & \text{for safety-critical.} \quad (16b) \end{cases}$$

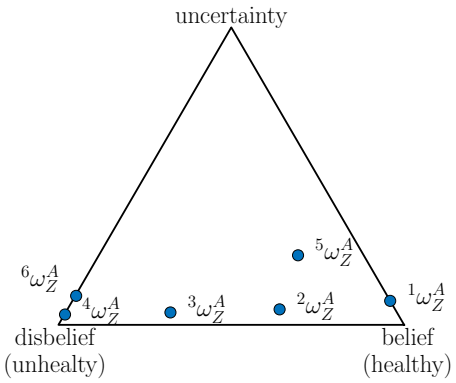
Using the CBF operator in Eq. (16a) provides the *overall state of health* formula. When, instead, co-multiplication is used in Eq. (16b), it provides the *safety-critical check*. The general behavior of these two functions is investigated, and their progressions are compared in our evaluation, presented in the next section.

# 6 Evaluation

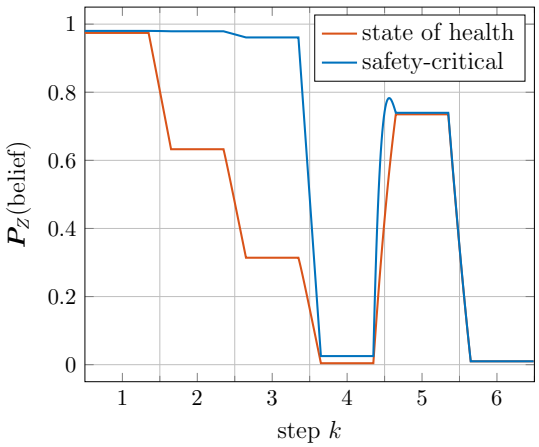
In this section, the two SA decision formulas proposed in this work (the *overall state of health* in Eq. (16a) and the *safety-critical check* in Eq. (16b)) are evaluated on the exemplary lucid system architecture introduced in Fig. 4. Using a fixed set of SL opinions as SA scores allows for a clear structure and investigation. However, our proposed method is applicable to any SL input opinions.



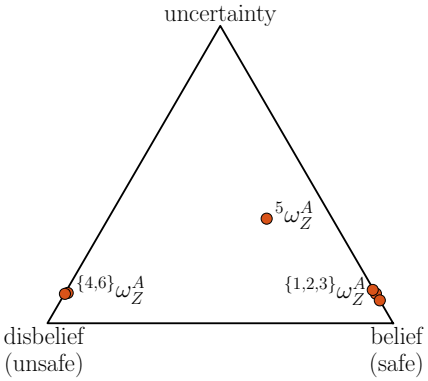

**(a)** Self-assessment input opinions.




**(c)** Overall state of health.




**(b)** Projected probability progression.




**(d)** Safety-critical check.

**Fig. 7.** Illustration of the evaluation scenario and its results. In (a), the SA input opinions are depicted for each step in a tabular form. Each colored square represents a predefined opinion for most likely true (green), somewhat true (yellow), and most likely false (red), cf. Fig. 2b. These opinions are given by the SAs on module or sub-system level and are the input for our unified SA framework. Below in (b), the projected probability progression is plotted, and the steps are spatially aligned with the tabular view in (a). Intermediate input opinions are linearly interpolated to yield a smooth transition between the configurations. On the right in (c) and (d), the *overall state of health* and the *safety-critical check* results are depicted in a barycentric triangle, respectively. For readability reasons, the probability projection has been omitted in (c) and (d); it is available in (b) at the respective step.

To highlight specific behaviors of the proposed formulas, we defined six different SA situations of the system and then connected these situations in six successive steps. Each step contains our system's SA situation, which consists of the SA score for each SA module. These scores are the input to our unified SA framework. The input SA opinions for each step are listed in tabular form in Fig. 7a. A green square represents an opinion that is most likely true, a red square denotes an opinion that is most likely false, and a yellow is a somewhat true and rather uncertain opinion. These classified exemplarily opinions were introduced in Fig. 2b. For readability reasons, the sensor and processing SA scores are combined in this section, and the respective opinions are denoted by the variables $V_1$, $V_2$, and $V_3$.

Now, to investigate and compare the behavior of the two formulas, the predefined SA input opinions slowly fade in and out step by step. Intermediate opinions are calculated between two steps to provide a smooth transition between the different SA input situations. For each set of intermediate opinions, the unified SA opinions are obtained using our two formulas (Eq. (16a) & (16b)). The progression of these opinions' projected probability is plotted in Fig. 7d. At each of the defined steps, the obtained opinions are illustrated in Fig. 7b for the *overall state of health* and in Fig. 7c the *safety-critical check*.

In general, when all SA input opinions indicate healthy states and no issue is reported (cf. Step 1), the two formulas provide similar scores. However, as soon as some modules start to fail and the SA input opinions report this, the results differ. In Step 2, the first sensor and processing (denoted by $V_1$) report issues, followed by the concurrent SA (denoted by $O$). Since there is still a sufficient number of sensors available for fusion, the *safety-critical check* is hardly affected. Contrarily, the *overall state of health* shows a clear decrease. This is the same for Step 3, where the second sensor and processing part (denoted by $V_2$) fails. The *safety-critical check* shows that enough modules are still working well to ensure safe operation. Comparing it with the *overall state of health*, their difference reveals that there are already some issues, and the margin of redundancy shrank. As soon as the last sensor (denoted by $V_3$) fails in Step 4, both formulas report similar results, saying that the system state is unhealthy and unsafe. Steps 1 to 4 demonstrate the applicability of our proposed solution to parallel and redundant parts within the software stack, even when only observing the final output of two formulas. In Fig. 7b & 7c, the difference between the two formulas in switching from good to bad is emphasized. While the *overall state of health* transitions step by step, the critical check jumps harshly.

To highlight the advantages of using SL in this context, Step 5 shows uncertain SA information from the fusion and the planning & control modules. Although uncertain, it tends to be a true statement (somewhat true as visualized in yellow in Fig. 2b). As a result, the obtained unified SA opinions become uncertain as well (cf. ${}^{5}\omega_Z^A$ in Fig. 7b & 7c). Thus, the degree to which an SA module is certain about its statement is effectively being considered, and it is reflected by the end result. The small 'overshoot' when transitioning from Step 4 to 5 is due to the fairly simple linear interpolation of all variables. In this specific case, four modules change from unhealthy back to healthy while becoming uncertain; the combination of all in our formula results in non-linear behavior.

In both formulas, the presence of a working fusion and planning & control is considered mandatory. Consequently, when the fusion fails in Step 6, both yield a result telling the system that it is performing poorly. In contrast to before, no margin is available, and thus, the *overall state of health* changes as abruptly as the *safety-critical check*.

Overall, our evaluation considers multiple different system states, and the output of our formulas meets the expectations they have been created for. Thus, only considering the output of our proposed formulas provides insights into many aspects of the system.

# 7 Conclusion

Summarizing our work, we identified self-assessment (SA) as a key aspect to enable the safe and robust operation of automotive vehicles. First, we investigated the state-of-the-art SA approaches for autonomous driving and categorized them into SA on module, sub-system, and system levels. After providing an introduction to subjective logic (SL),

we discussed why it is a well-suited common interface as the basis for a system-wide framework investigating the SA state of a software stack. For an exemplary software stack architecture, we derived a mathematical representation using subjective networks (SNs) and elaborated typical patterns to allow adapting our method to any system structures in general. Investigating the proposed unified SA framework results for several system SA input configurations in our evaluations demonstrated the effectiveness of our approach, being able to verify both an *overall state of health* and a *safety-critical check*. By providing a first approach to fusion SA from different modules across the complete software stack, we aim to contribute to the robust and safe operation of autonomous vehicles.

# Ackowledgement

Parts of this work were supported by the State Ministry of Economic Affairs, Labour and Tourism Baden-Württemberg (project U-Shift II, AZ 3-433.62-DLR/60). Parts of this research have been conducted as part of the EVENTS project, which is funded by the European Union under grant agreement No. 101069614. Views and opinions expressed are, however, those of the authors only and do not necessarily reflect those of the European Union or European Commission. Neither the European Union nor the granting authority can be held responsible for them.

# References

[1] International Organization for Standardization. "ISO/PAS 21448: Road Vehicles — Safety of the Intended Functionality". In: *ISO, Publicly Available Specification* (2019).

[2] Thomas Griebel et al. "Kalman filter meets subjective logic: A self-assessing kalman filter using subjective logic". In: *23rd International Conference on Information Fusion (FUSION)*. 2020. DOI: 10.23919/FUSION45008.2020.9190520.

[3] Thomas Griebel et al. "Online Performance Assessment of Multi-Sensor Kalman Filters Based on Subjective Logic". In: *International Conference on Information Fusion (FUSION)*. 2023. DOI: 10.23919/FUSION52260.2023.10224188.

[4] Johannes Müller, Michael Gabb, and Michael Buchholz. "A Subjective-Logic-based Reliability Estimation Mechanism for Cooperative Information with Application to IV's Safety". In: *IEEE Intelligent Vehicles Symposium (IV)*. 2019, pp. 1940–1946. DOI: 10.1109/IVS.2019.8814153.

[5] Johannes Müller et al. "A Trust Management and Misbehaviour Detection Mechanism for Multi-Agent Systems and its Application to Intelligent Transportation Systems". In: *IEEE 15th International Conference on Control and Automation (ICCA)*. 2019, pp. 325–331. DOI: 10.1109/ICCA.2019.8899968.

[6] David M. Clarke and Ian Hollister. "Introduction to Redundancy". In: *Safety and Reliability* 30.4 (2010), pp. 4–15. DOI: 10.1080/09617353.2010.11690919.

[7] Rens W. Van Der Heijden et al. "Survey on misbehavior detection in cooperative intelligent transportation systems". In: *IEEE Communications Surveys and Tutorials* 21 (2019), pp. 779–811. DOI: 10.1109/COMST.2018.2873088.

[8] Michael Wolf et al. "Securing CACC: Strategies for Mitigating Data Injection Attacks". In: *IEEE Vehicular Networking Conference, VNC*. Vol. 2020-December. IEEE Computer Society, 2020. DOI: 10.1109/VNC51378.2020.9318396.

[9] Thomas Griebel et al. "Self-Assessment for Single-Object Tracking in Clutter Using Subjective Logic". In: *International Conference on Information Fusion (FUSION)*. 2022. DOI: 10.23919/FUSION49751.2022.9841294.

[10] Thomas Griebel et al. “Self-Assessment for Multi-Object Tracking Based on Subjective Logic”. In: *IEEE Intelligent Vehicles Symposium (IV)*. 2024, pp. 1750–1757. DOI: 10.1109/IV55156.2024.10588720.

[11] Oliver Schumann et al. “Self-Assessment of Evidential Grid Map Fusion for Robust Motion Planning”. In: *IEEE International Conference on Intelligent Transportation Systems (ITSC)*. 2024. DOI: 10.48550/arXiv.2409.20286.

[12] Yaakov Bar-Shalom and Thomas E. Fortmann. *Tracking and Data Association*. Academic Press, New York, 1988.

[13] Anne Stockem Novo et al. “Self-evaluation of automated vehicles based on physics, state-of-the-art motion prediction and user experience”. In: *Scientific Reports* 13.1 (Aug. 2023), p. 12692. DOI: 10.1038/s41598-023-39811-1.

[14] Thomas Wodtko, Alexander Scheible, and Michael Buchholz. “Self-Assessment and Correction of Sensor Synchronization”. In: *IEEE 27th International Conference on Intelligent Transportation Systems (ITSC)*. 2024. DOI: 10.48550/arXiv.2409.20266.

[15] Bernhard Hafner et al. “A Survey on Cooperative Architectures and Maneuvers for Connected and Automated Vehicles”. In: *IEEE Communications Surveys and Tutorials* 24.1 (2022), pp. 380–403. DOI: 10.1109/COMST.2021.3138275.

[16] Francesco Amato et al. “A direct/functional redundancy scheme for fault detection and isolation on an aircraft”. In: *Aerospace Science and Technology* 10.4 (2006), pp. 338–345. DOI: 10.1016/j.ast.2006.03.002.

[17] Matthijs Klomp et al. “Trends in vehicle motion control for automated driving on public roads”. In: *Vehicle System Dynamics* 57.7 (2019), pp. 1028–1061. DOI: 10.1080/00423114.2019.1610182.

[18] Thomas Wodtko et al. “Conflict Handling in Time-Dependent Subjective Networks”. In: *27th International Conference on Information Fusion (FUSION)*. 2024, pp. 1–8. DOI: 10.23919/FUSION59988.2024.10706464.

[19] Audun Jøsang. *Subjective Logic: A Formalism for Reasoning Under Uncertainty*. Springer International Publishing, 2016. DOI: 10.1007/978-3-319-42337-1.

[20] Audun Jøsang, Jie Zhang, and Dongxia Wang. “Multi-source trust revision”. In: *International Conference on Information Fusion (FUSION)*. 2017. DOI: 10.23919/ICIF.2017.8009635.

[21] Rens W. van der Heijden, Henning Kopp, and Frank Kargl. “Multi-Source Fusion Operations in Subjective Logic”. In: *International Conference on Information Fusion (FUSION)*. 2018. DOI: 10.23919/ICIF.2018.8455615.