
Proceedings Paper:
Burton, SIMON, Shahbeigi Roudposhti, Sepeedeh and Zou, Jie (2026) Effective and reflective assurance for AI-based autonomy. In: 34th Safety-Critical Systems Symposium. Safety Critical Systems Club.

Effective and reflective assurance for AI-based autonomy

Simon Burton, Jie Zou, Sepeedeh Shahbeigi Roudposhti

Centre for Assuring Autonomy, University of York, York , UK

Abstract *This paper outlines the challenges in assuring the safety of autonomous AI-based systems and the ensuing gaps in assurance that need to be closed to achieve a comparable level of integrity as for conventional safety-critical systems. By analysing the underlying causes of uncertainty in assurance arguments for these systems, as well as current research on this topic, the paper identifies extensions to conventional safety assurance approaches, that, if combined within a holistic framework, may go some way to reducing these gaps. As part of this approach, we recommend the use of causal models of the environment and system that enable more rigorous statements to be made about the completeness and consistency of safety requirements and for supporting verification and validation activities. Nevertheless, some assurance uncertainty will inevitably remain as an inherent consequence of the environmental and system complexity. Therefore, a reflective and iterative approach to assurance is deemed essential to create transparency regarding the strength or otherwise of the safety claims that can be made about the system.*

1 Introduction

Artificial Intelligence (AI) and specifically Machine Learning (ML) technologies are increasingly being used to enable autonomous operations of cyber-physical systems including automated driving, factory automation, and autonomous sea vessels. With sufficient training data, an ML-based system can operate within otherwise poorly specified and evolving environments. However, whilst much effort has been invested in recent years, large-scale deployment in these sectors has been slower than initially anticipated amidst high-profile accidents and regulatory uncertainty. In addition, established approaches to safety assurance, as defined in safety regulations and standards, do not transfer easily to the development and test approaches used in the ML community.

This paper provides an examination of these assurance challenges and summarises recent work in extending systems safety engineering approaches to

support AI-based autonomous systems operating in complex environments. In doing so, we provide an optimistic outlook for how progress towards convincing safety assurance arguments for AI-based autonomy is being made. The paper begins with a comparison of traditional systems safety engineering approaches with the challenges and current practice in assuring ML-based autonomy. Section 3 examines these challenges through the lens of the root causes of assurance uncertainty associated with such systems. Based on principles of effective assurance arguments, Section 4 outlines extensions to established safety assurance approaches currently being developed to fill these assurance gaps. The paper illustrates the proposed assurance principles based on the example of automated driving. However, we believe the same principles apply also to other applications of AI-based autonomy such as autonomous maritime vessels and industrial robotics.

2. Software safety assurance - then and now

2.1. Conventional software safety assurance

Systems safety engineering processes determine risks associated with hazardous (mal)functioning of the system and then implement, verify, and validate measures for avoiding or mitigating hazardous events caused by technical system failures. In doing so, potential contributions of hardware and software components to both the causes of hazards as well as the implementation of mitigation measures are defined. Safety cases and assurance arguments have become a common means for demonstrating that a sufficient level of safety has been achieved in the system prior to its deployment and continuing during its operation and are required by many safety standards. ISO 15026-1:2019[1] defines an assurance argument as:

> "a reasoned, auditable artefact that supports the contention that its top-level claim (or set of claims) is satisfied, including systematic arguments and its underlying evidence and explicit assumptions that support the claim(s)"

Assurance arguments are often supported by graphical notations such as GSN[2] and provide a means of communicating with stakeholders of the system including regulatory and approval bodies and customers.

Over the last 40 years, the safety assurance of software-based cyber-physical systems has matured into an established engineering discipline, with best practices widely adopted within industry and encoded in foundational standards such as IEC

[1] ISO 15026 Systems and software engineering — Systems and software assurance

[2] Goal Structuring Notation (GSN) standard (version 3)

61508[3] as well as sector-specific guidance such as DO-178C[4] (airborne systems and equipment) and the ISO 26262[5] (road vehicles). (Hawkins, Habli, & Kelly, 2013) summarise the core principles of such standards as shown in Table 1, which also lists typical measures required by the standards related to each principle:

Table 1: Software safety assurance principles and typical measures required by standards

Assurance principles	Typical measures in software development processes
1 - Software safety requirements shall be defined to address software contribution to system hazards	Detailed definition of software safety requirements, traceable from system level down to individual code units.
2 - Intent of software safety requirements shall be maintained throughout requirements decomposition	Software architecture design based on principles such as low coupling and strong cohesion, use of pre-qualified components, freedom from interference mechanisms, programming guidelines supported by automated analysis and peer review.
3 - Software safety requirements shall be satisfied	Unit testing including structural code coverage and test case elicitation against detailed specifications, verification of software code against formal specification, software integration testing and hardware software integration testing, system level validation including failure conditions.
4 - Hazardous behaviour of software has been identified & mitigated	Software safety analysis to determine potentially hazardous effects of code defects and to determine appropriate process and architectural counter measures, static code analysis to demonstrate freedom from run-time errors such as divide-by-zero, as well as properties such as worst-case execution time, memory consumption, etc.
4+1. Confidence established in addressing software safety principles shall be commensurate to contribution of software to system risk	Selection of architectural and development measures based on safety integrity levels (e.g. DALs or SILs), development of a software safety assurance argument (safety case), process audits, safety reviews and assessments.

These principles declare that safety is first defined as a system-level property and then refined into a set of traceable requirements at the level of source code. Safety is "designed into" the software through a series of design and process measures. These include rigorous planning, architectural and coding guidelines and the static analysis of critical properties of code. In addition, a verification and validation strategy based on a detailed understanding of the types of faults that can be introduced at various stages of the development process is developed. This ensures

[3] IEC 61508 Functional safety of electrical/electronic/programmable electronic safety-related systems

[4] DO-178-C Software Considerations in Airborne Systems and Equipment Certification

[5] ISO 26262 Road vehicles - functional safety

a systematic approach to demonstrating properties of the software, using a combination of complementary design, analysis, review, and test methods.

2.2 Current limitations of the assurance of ML-based autonomy

With the emergence of AI-based autonomy and, in particular, automated driving systems (ADS), we have seen the emergence of a new set of safety principles. The ISO 21448[6] *Safety of the Intended Functionality* (SOTIF) standard extends the functional safety requirements of ISO 26262 by addressing the risk due to hazards resulting from functional insufficiencies of the intended functionality and reasonably foreseeable misuse. ISO 21448 emphasizes the definition of an Operational Design Domain (ODD), the identification of potential performance insufficiencies and vehicle-level strategies to compensate for these insufficiencies. Unlike ISO 26262, the standard allocates quantitative acceptance criteria to errors that are not directly caused by random hardware faults. These are conceptualised as conditional probabilities based on latent insufficiencies of the system and the occurrence of triggering conditions that reveal these. Applying SOTIF principles for AI-based systems introduces several challenges. These include translating risk acceptance principles into quantitative thresholds, tracing these to measurable properties of the ML models, and managing requirements on datasets. These topics are addressed in ISO PAS 8800[7] *Road Vehicles - Safety and Artificial Intelligence*.

Demonstrating the satisfaction of SOTIF requirements in ADS is currently strongly reliant on statistical validation. However, there is a fundamental infeasibility of physically accumulating the necessary billions of miles of collision-free driving required to demonstrate statistical relevance for target failure rates. Three interconnected issues exacerbate the problem of reliance on test evidence alone: *controllability* of the precise conditions that could lead to failures is severely restricted during physical testing, making it difficult to precisely and consistently manipulate all relevant attributes of the environment and vehicle state needed to rigorously test the system against known triggering events; *observability* is compromised by the opaque nature of the AI algorithms leading to a lack of explainability of system behaviour; and *repeatability* is fundamentally undermined by aleatoric uncertainty (domain randomness) and epistemic uncertainty (sensor errors). Current safety standards do not recommend the use of online learning, whereby each individual instantiation of a function deployed in the field could be adapted over time, e.g. through a process of reinforcement. Such approaches would further undermine the repeatability of test cases and the continuing validity of an assurance argument. Although simulation aids in achieving controllability and repeatability, proving the transferability of those simulated results to the real world remains a challenge. Furthermore, the increased effort required to assure AI-based

[6] ISO 21448 Road vehicles — Safety of the intended functionality

[7] ISO PAS 8800 Road Vehicles — Safety and Artificial Intelligence

autonomous systems must be repeated for each update of the system (e.g. via over-the-air updates to compensate for changes in the ODD over time).

Reflecting again on the 4+1 principles of (Hawkins, Habli, & Kelly, 2013), the challenges associated with assuring the safety of ML-based autonomous functions operating in complex environments are summarised in Table 2. Whilst achieving functional safety standards such as ISO 26262 is still a pre-requisite, current assurance approaches for demonstrating the safety of complex autonomous and AI-based systems do not satisfy the well-established principles of safety assurance. We therefore see a pressing need to address these shortcomings.

Table 2: Gaps in assurance principles for AI-based autonomy

Assurance principles	**Challenges and gaps in current practice**
1 - Software safety requirements shall be defined to address software contribution to system hazards	Safety requirements are defined based on scenario descriptions and quantitative acceptance criteria, referencing taxonomies describing relevant aspects of the ODD. However, these requirements lack specificity and fail to address the full range of potential hazards.
2 - Intent of software safety requirements shall be maintained throughout requirements decomposition	Semantic gaps caused by the complexity of the environment, task and data-driven approach to the specific result in incomplete technical specifications that do not cover the full (often implicitly stated) intent of the systems designers. High-level risk acceptance criteria are not directly traceable to ML performance metrics.
3 - Software safety requirements shall be satisfied	The satisfaction of safety requirements is primarily validated through statistical methods. These methods do not scale to extremely low target failure rates, critical but rare triggering conditions and continual changes to the system or the environment. Invalid assumptions regarding the representativeness of test data, the distribution of failures within the input space and the integrity of the test data itself undermine the contribution of this evidence.
4 - Hazardous behaviour of software has been identified & mitigated	Residual errors in ML models will inevitably remain. An understanding of the impact of these errors on system level behaviour depends on the properties of the input space that can trigger errors in the model as well as the distribution of these triggering conditions within the operating domain, both of which are non-trivial to characterise.
4+1. Confidence established in addressing software safety principles shall be commensurate to contribution of software to system risk	Compared to established approaches, there can be a significant lack of confidence in the results of the assurance activities for AI-based autonomous systems operating in complex environments, due to the shortcomings addressed above in principles 1 to 4. A level of assurance confidence, commensurate to the system risk requires additional measures.

3. Understanding the fundamental causes of assurance uncertainty

3.1. Complexity, Uncertainty and Semantic Gaps

From the perspective of complexity science (Erdi, 2007), complex systems share characteristics including: semi-permeable system boundaries, non-linear behaviour, mode transitions & tipping points, and self-organisation. For the purposes of this paper, we use the following definition of a complex system:

> **Complex system:** A system that exhibits behaviours that are emergent properties of the interactions between the parts of the system, where the behaviours would not be predicted based on knowledge of the parts and their interactions alone.

Such properties undermine typical approaches to safety assurance which are based on **models of known system behaviour and causes of failures** due to individual component faults. The lack of knowledge about the causes of emergent behaviour that is at the core of the definition of complexity used above is strongly related to the concept of uncertainty as illustrated in the following definition (Walker, et al., 2003):

> **Uncertainty:** Any deviation from the unachievable ideal of completely deterministic knowledge of the relevant system.

In cyber-physical systems operating in open and dynamic environments, various manifestations of uncertainty resulting from complexity can be illustrated as follows, inspired by (Lovell, 1995):

- The **environment** (e.g., urban traffic) is naturally complex, unpredictable and changing. Interactions between actors in the environment can lead to behaviour that could not have been predicted at design time.
- **Observations** of the environment are made through imperfect **sensors**, which are subject to inherent limitations such as restricted resolution, field of view, and signal noise. Not all relevant properties of the environment can be directly observed and may need to be inferred via other channels.
- The system processes these observations to determine appropriate actions using a blend of **algorithms**, **heuristics**, and **ML** techniques. These methods often rely on models with inherent epistemic uncertainty.

The relationship between system complexity and uncertainty lies at the core of many of the challenges related to the assurance of autonomous AI-based systems (Burton, McDermid, Garnet, & Weaver, 2021) and leads to the following definition of assurance uncertainty:

> **Assurance uncertainty:** a lack of knowledge—and consequently, a lack of confidence—regarding the completeness and/or validity of an assurance argument for the system's critical properties.

ISO 21448 defines functional insufficiencies as arising from two main sources: specification insufficiencies and performance insufficiencies. Both can be interpreted within the framework of the previously described uncertainty model. These insufficiencies represent specific manifestations of uncertainty, which ultimately contribute to uncertainty in the assurance argument. Specification insufficiencies are related to the validity and completeness of appropriate safety acceptance criteria and the definition of acceptably safe behaviour in all situations that can reasonably be anticipated to arise within the target environment. Specification insufficiencies can also be rooted in competing objectives and stakeholder-specific definitions of acceptable residual risk leading to unresolved questions related to ethical/socially acceptable system behaviour. The inability to provide a complete specification of the (safe) behaviour of the system as well as the input space of ML components is inherently linked to both the **semantic gap** (Burton, et al., 2020) and emergent properties of complex systems. Performance insufficiencies relate to a lack of predictability in the performance of the technical system components. An example of performance insufficiencies in ML models are the unpredictable reaction of a system to previously unseen events (lack of generalization), or differences in the system behaviour despite similar input conditions (lack of robustness). The relationship between environmental, observational, system complexity and assurance uncertainty is summarised in **Fig. 1** (Burton & Herd, 2023).

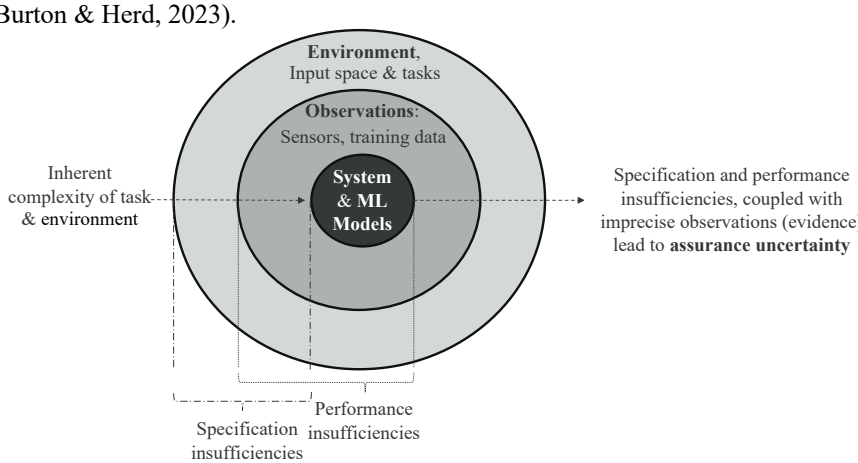

Fig. 1: Relationship between system complexity and assurance uncertainty

A recently published paper (Porter, et al., 2025) proposed a classification framework (INSYTE) for analysing various dimensions of AI-based and Autonomous Systems. INSYTE considers the essential characteristics of an AI system across eight key dimensions grouped into four categories:

- System design (under-specification and adaptiveness);
- Functionality (breadth and depth);
- Operating environment (diversity and dynamism); and

- Independence from human operational control (intervention and oversight).

These categories could be used as the basis for an evaluation of potential root causes of assurance uncertainty within a system thus providing qualitative criteria to determine their "assurability" given state-of-the-art assurance methods.

4. Closing the gaps

4.1 Effective safety assurance arguments

The previous sections have highlighted the inherent uncertainties associated with AI-based autonomous systems which undermine confidence in safety assurance arguments. However, emerging solutions are beginning to bridge these gaps, which we summarise in this section. We structure this analysis according to the following properties of effective assurance arguments[8] and argue that, by striving to achieve these properties, assurance uncertainty can be reduced for autonomous ML-based systems.

4.2 Clear definition of the safety claim to be demonstrated

Conventionally, technical safety requirements on the system are derived from an analysis of the potential harm to operators, bystanders and the environment caused by technical failures in the system. This leads to a detailed definition of requirements that are achieved through constructive system design and operational measures and confirmed through a combination of analysis and testing. However, for autonomous AI-based systems, the complexity and unpredictability of both the environment and the system itself, coupled with the transfer of decision-making responsibility from humans to the systems require going beyond purely technology-centric assurance arguments (Burton, et al., 2020). Nevertheless, without a comprehensive and explicit specification, it is not possible to conclusively demonstrate that safety requirements of the system are met, leading to assurance uncertainty and the lack of a social and regulatory acceptance of the system.

Standards such as ISO 21448 recommend applying principles such as ALARP, GAMAB, MEM[9] and positive risk balance to determine risk acceptance criteria for

[8] Inspired by Natarajan Shankar, SRI, in his Keynote at SAFECOMP, 2023.

[9] ALARP: As Low As Reasonably Practicable, GAMAB: Globalement Au Moins Aussi Bon (French for globally at least as good), MEM: Minimum Endogenous Mortality

the systems. However, these approaches do not address the subtleties of operating in complex socio-technical contexts and ethical considerations of such systems such as the equitable distribution of risk, human agency and transparency. An **ethics-informed analysis** (Porter, Habli, McDermid, & Kaas, 2024) must therefore be applied to determine levels of acceptable risk as well as supporting requirements on the system necessary to achieve ethical acceptability.

Hazards are not only the result of technical product failures but can be caused by unintended and unforeseen interactions between the system and its operating context. It must also be acknowledged that insufficiencies in the technical system and residual assurance uncertainty will inevitably remain. Therefore, there is a strong need for **operational safety concepts** and associated assurance arguments. Erz et al. (Erz, Burton, & Sax, 2025) argue that the ISO 26262 and ISO 21448 safety life cycle and assurance approaches should be explicitly extended to include the safety of driving policy specification and safety in use (referred to as "*operational safety*"). This includes dedicated measures for field data evaluation, risk analysis and counter-measure definition. The Guidance on the safety assurance of autonomous systems in complex environments (SACE) (Hawkins, et al., 2022) describes an assurance framework that addresses the assurance of an autonomous within its wider operational context. This includes the development of an **operational domain model** that is used as a basis of an operational hazard analysis and the definition of an operational safety concept. Furthermore, safety requirements are derived to manage residual technical failures of the system and to monitor whether the system is operating outside of its defined set of operational parameters.

In parallel with these developments, recent research has focused on improving the **traceability between SOTIF-driven safety intentions and the safety requirements allocated to ML components**. A recurring concern in the literature is that, while SOTIF identifies performance insufficiencies and contextual limitations as central contributors to hazardous behaviour, it provides limited guidance on how such insights should be translated into explicit, verifiable constraints for data-driven perception systems (Burton, Hellert, Hüger, & Mock, 2022). This has motivated work on structured decomposition frameworks that seek to bridge the gap between high-level operational safety objectives and the behavioural expectations placed on ML components operating within the defined ODD (Roudposhti, et al., 2025). By grounding requirements elicitation and refinement in the operational domain model, these approaches ensure that the intent expressed at the system level is preserved as it is propagated down to ML-based functions. This line of work also emphasises the importance of demonstrating that the behaviour of ML components is not only aligned with SOTIF requirements under nominal conditions but remains sufficiently robust across the operational variations that give rise to potential insufficiencies. To this end, ML-level requirements explicitly account for contextual sensitivity and uncertainty. Furthermore, dataset requirements, relating to relevance, coverage and quality, must reflect the operational conditions under which safety must be assured. This increase in traceability provides the basis for more transparent and defensible assurance

arguments, helping to close the semantic gap between intended system functionality and the explicit specifications needed to assure ML-enabled perception in complex autonomous systems.

4.2 Arguments based on rigorous models of system & its context

Early efforts to define ODDs relied predominantly on *informal* or *semi-structured* descriptions of the environmental and operational context. These offer practical utility but lack the semantic precision required for assessing autonomy behaviour under complex and evolving contexts. This informality leads to several limitations: ambiguity in attribute definitions, difficulty in establishing traceable behavioural guarantees, and the absence of systematic mechanisms for relating contextual factors to system-level safety requirements. We therefore see the need for *rigorous* and *formally grounded* representations of both the system and its operating context.

Recent efforts have moved toward structured ODD specifications. Standards such as ISO 34503[10] and ASAM OpenODD[11] introduce hierarchical attribute models and machine-readable schemas, improving consistency and interoperability. However, they remain primarily descriptive and do not define behavioural semantics linking ODD attributes to system behaviour (e.g. road surface friction and braking distance). As a result, structured ODDs alone are insufficient for reasoning about safety, performance degradation, or context-dependent behaviour.

The SOCA (Situation–Object–Context–Action) framework introduced by (Butz, et al., 2020) formalises the structure of operating context. It decomposes driving scenarios into layered abstractions and maps them to system actions, providing an ontology that improves consistency across engineering workflows. However, SOCA remains largely static and does not model uncertainty, latent variables, or causal relationships, limiting its ability to explain how context evolution affects system safety. The Context-Based and Bayesian Network (CBN) approach proposed by (Gansch, Putze, Koopmann, Reich, & Neurohr, 2025) extends structural models by introducing probabilistic representations of context. ODD-relevant factors such as weather, friction, and illumination are modelled as random variables, enabling uncertainty-aware assessment and inference under partial observability. Nevertheless, the model remains primarily correlational: causal relationships are implicit, interventions are unsupported, and behavioural effects are encoded indirectly, limiting its suitability for safety assurance and control reasoning.

[10] ISO 34503 Road Vehicles – Test scenarios for automated driving systems – Specification for operational design domain

[11] ASAM OpenODD modelling approach and exchange format

(Sifakis & Harel, 2023) argue that correctness in autonomous systems requires integrating learning-based components with formal models as data-driven systems lack guarantees, while model-based systems lack scalability and adaptability. They advocate architectures in which learning is constrained by semantic models, and system behaviour is defined relative to explicit environmental assumptions.

Building on these insights, we propose the use of Structural Causal World Models (SCWMs) that provide a unified, causally grounded representation of both system behaviour and operational context (Zou, et al., 2026). SCWMs advance the state of the art by:

- Representing ODD attributes as causally linked variables rather than static descriptors;
- Combining symbolic specification, probabilistic uncertainty, and causal semantics;
- Enabling inference over latent context and belief updating at runtime; and
- Supporting formal arguments for ODD conformance and safety validity.

SCWMs therefore integrate the structural clarity of SOCA, the uncertainty modelling of CBN, and the correctness principles of hybrid ML–model-based control, providing a rigorous foundation for context-aware autonomy and runtime assurance.

4.3 Evidence and arguments that can be easily refuted or believed

Establishing statistically relevant statements about critical failures—such as arguing the mean distance between collisions of 3.85 million km (based on German crash statistics) with a 95% confidence level — requires driving 11.6 million test kilometres without collision if using statistical validation alone (Åsljung, Joans, & Jonas Fredriksson, 2017). To address this challenge, methodologies have been developed that combine data analysis with robust system design arguments. Extreme Value Theory (EVT) (Songchitruksa & Tarko, 2006), (Jonasson & Rootzen, 2014) proposes a statistical solution by modelling extreme events (near-misses or critical situations) using a statistical distribution, which is then extrapolated to deduce the probability of safety goals being violated. Nevertheless, doubts may persist about the validity of assumptions underlying the statistical analysis that could undermine the results such as the equal distribution of triggering events within the environment and the representativeness of the test scenarios.

Work by BMW (Werling, Faller, Betz, & Straub, 2025) reduces the reliance on system level statistical testing by grounding assurance in comprehensive system design. Their approach employs a **combination of systems engineering, risk analysis, data analysis, and statistical learning** to quantify uncertainties associated with hazard scenarios within a redundantly designed system. This framework propagates component-level uncertainties using Bayesian Networks,

demonstrating that system design—such as a redundant 2-out-of-3 voting architecture—can reduce the necessary validation data by factors of 100 or more, thereby quantifying risk through stochastic simulation. The approach enables an iterative validation loop that guides developers on whether to focus on implementing system improvements or collecting more data to meet the meet residual risk acceptance criteria.

Subjective Logic (SL) (Josang, 2016) allows for reasoning with uncertain beliefs, combining probabilistic logic and evidence theory. (Burton & Herd, 2024) have used SL to reason about the **uncertainty associated with the use of ML metrics as safety evidence**. In their approach, the claim of the argument is formulated as safety contract with a guarantee (property to be demonstrated), quantitative target, assumptions, and level of (statistical) confidence with which the claim should be demonstrated. Primary evidence, such as test results, is gathered and expressed as an opinion in SL. This evidence is then evaluated for uncertainty, and adjusted using secondary evidence, such as data labelling quality or representativeness. The resulting opinion is assessed against the safety contract, and if necessary complemented with additional evidence that increases the belief in the presented results. The residual uncertainty can be used to identify a conservative estimate of the property being measured within certain confidence bounds.

Beyond the validity of individual pieces of evidence, the overall structure and underlying assumptions of the assurance must also be scrutinised to address assurance uncertainty (Hawkins, Kelly, Knight, & Graydon, 2011) (Burton & Herd, 2023). Waymo (Favaro, et al., 2023) defines safety as the Absence of Unreasonable Risk (AUR) and structures its case around three perspectives: a layered decomposition of hazards, an iterative safety evaluation lifecycle, and a **credibility assessment of both evidence and arguments**. A distinctive feature of Waymo's approach is its attempt to counter confirmation bias through **dialectic arguments**. They employ a tabular Claims, Arguments, Evidence (CAE) structure where statements of limitations, and counter-arguments are expressed. By requiring that alternatives be articulated and rejected within the safety case itself, Waymo frames its safety case as a dialectic exercise where claims are not only supported but also pressure-tested against possible objections. Similarly, the Assurance 2.0 framework (Varadarajan, et al., 2023) focuses on the validity and soundness of argumentation, the use of reusable theories, and the systematic exploration and resolution of defeaters to ensure that completed assurance cases are indefeasible.

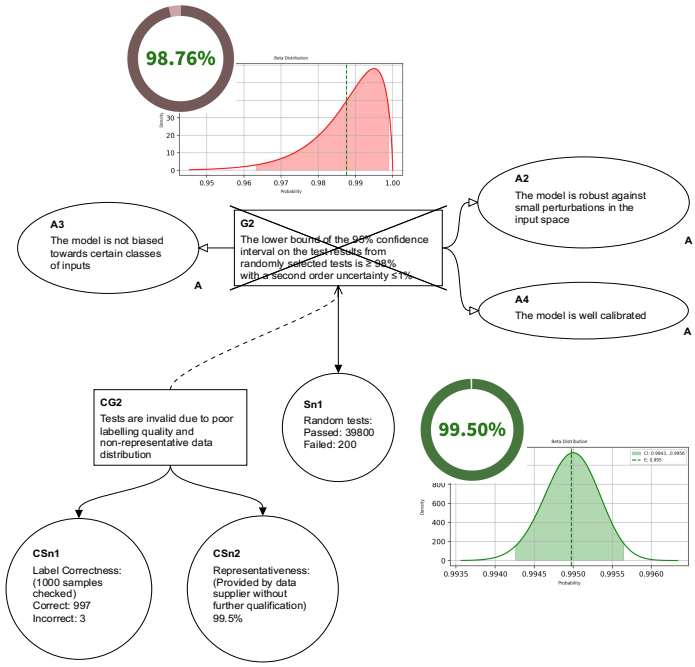

Fig. 2: Dialectic evaluation of a claim regarding ML performance (GSN Excerpt)

In combination, the techniques mentioned in this section can be used to identify, quantify and reduce assurance uncertainty, such that it can be explicitly considered when making deployment decisions. Although existing safety standards require assurance arguments to be the subject of rigorous independent review, there is a presumption that processes outlined in the standards, if followed will lead to suitably convincing arguments. We argue, that as system complexity increases, this is no longer sufficient, and additional **reflective approaches to safety assurance** such as dialectics are required to strengthen safety cases for these systems. As an example of how these perspectives can be combined, **Fig. 2** uses the GSN dialectic extension to demonstrate how claims based on ML metrics can be questioned, based on measurable properties of the test data. In (Burton & Herd, 2024), SL was used to demonstrate how the primary evidence (measured at 99.5% recall) could be adjusted to provide a more conservative, uncertainty-aware estimation of the model performance. The analysis shows that although the adjusted recall rate (98.76%) is still above its target value, doubts regarding the integrity of the testing approach add uncertainty such that the lower bound of the 95% confidence interval is below the target (illustrated by the graphs depicting the probability distribution function of the collected evidence). The dialectic analysis in this example could be further extended to evaluate the impact of doubt in the assumptions listed in the argument (e.g. regarding model calibration or the quality of sensor inputs).

4.4 Assurance driven workflows and through-life safety

Functional safety standards take a traditional V-model view of a systems engineering process. Safety analyses are used to determine risk, potential failure modes and counter-measures. This leads to additional requirements at different levels of abstractions from system safety goals through to technical safety requirements on hardware and software components. These requirements are related via vertical traceability across system abstraction levels and their refinement supported by constructive measures during design and implementation. Verification and validation evidence is also linked directly to requirements at each level of refinement. This approach ensures that evidence collected during development and test can be clearly related to safety goals supporting the assurance argument.

ISO 21448 presents a more iterative approach whereby newly uncovered functional insufficiencies and triggering conditions discovered during testing and operation are used to update specification and design. Verification and validation is performed until the residual risk due to known and unknown scenarios is considered sufficiently small. By integrating a **continual process of field monitoring and validation during operation**, successive approaches to a controlled widening of the ODD and level of authority over the system can be achieved whilst bootstrapping statistical validation of the functionality. Although this approach is being used by major manufacturers and suppliers of automated vehicles, see e.g. (Werling, Faller, Betz, & Straub, 2025), specific methods for which system and environment properties should be monitored during operation and which statistical reasoning methods should be applied are yet to be encoded in standards. Furthermore, technical measures that execute at run-time such as monitoring for ODD exits, plausibility checks, and minimal risk manoeuvres are also recommended by standards such as ISO TS 5083 *Safety for automated driving systems*. We see the use of formalised structural causal world models of the ODD (as described in Section 4.2) as a mechanism by which **run-time monitors** could be directly synthesised in a verifiable manner, that could include the ability to indirectly infer properties which are not directly measurable by onboard sensors.

ISO PAS 8800 also proposes a highly iterative safety lifecycle that mimics the combination of DevOps and DataOps (also known as MLOps) approaches used by industry-scale ML developers. As development of the AI system progresses, ISO PAS 8800 foresees that different Key Performance Indicators (KPIs) will be used to steer the iterative development, beginning initially with broad performance metrics before focusing on directly safety-related evidence with high confidence directly before release. **Safety analyses** form the "engine room" of the iterative AI safety lifecycle of ISO PAS 8800, whereby errors found during the current iteration are analysed for their criticality, potential causes hypothesised and appropriate measures defined, the effectiveness of which are evaluated during the next iteration.

A common goal across both the system and ML component layers of continual assurance as described above, is that **residual uncertainties in the assurance argument are minimised over time**. This includes not only the discovery of triggering conditions and insufficiencies that were not considered during development (e.g. due to inadequate modelling of the ODD), but also those that occur due to changes in the environment or the operation of the system.

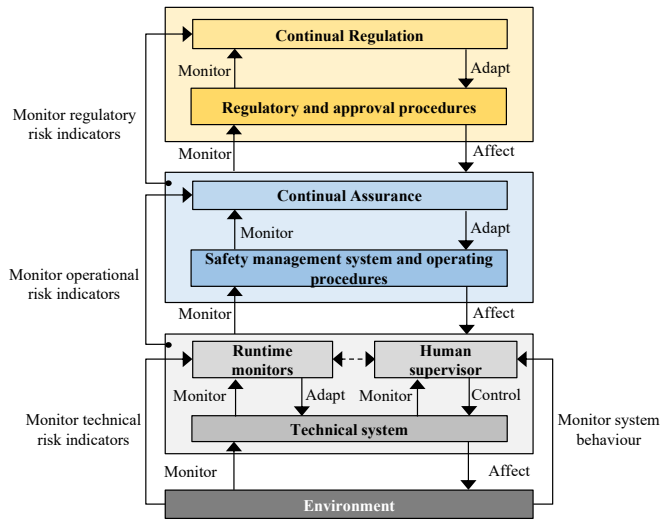

Fig. 3: Through-life assurance activities across technical, operational and governance socio-technical layers

Finally, as outlined in (Burton & McDermid, 2023) there is a need to embed through-life assurance activities within a process that not only monitors the technical performance of the system, but also the conditions of its safe usage to determine insufficiencies in operational safety concepts. Likewise, the deployment of these systems cannot wait until regulation has been passed to address all possible risks and eventualities. A bootstrapped approach as presented in **Fig. 3** is needed (and is already being followed in some countries[12]) that makes use of regulatory sandboxes that would accompany pilot projects to support the development and evaluation of approval procedures. A key challenge in this approach is the determination of leading operational and regulatory indicators that would effectively determine the conditions by which regulatory, approval and operational changes need to be made prior to unacceptable consequences of residual risk.

[12] https://www.europarl.europa.eu/RegData/etudes/BRIE/2022/733544/EPRS_BRI(2022)733544_EN.pdf

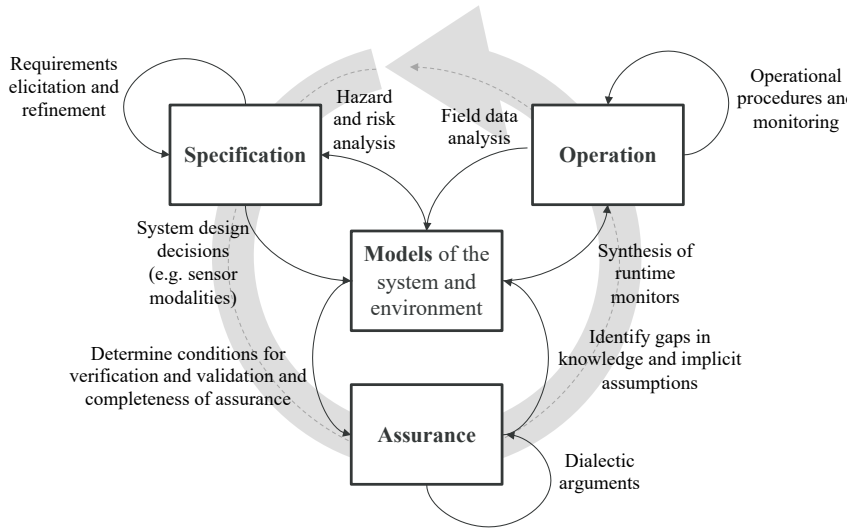

Figure 4: Summary of relationships within an iterative and reflective model-based assurance approach

5 Conclusions

In this paper, we have presented trends in safety assurance of autonomous AI-based systems that, taken together, mark a significant shift in systems safety engineering whilst still adhering to important key principles. In summary, we note that the use of ML does not bypass the need for detailed safety requirements, but arguing the completeness and consistency of these requirements becomes harder the with increasing autonomy, open context environments and more powerful AI approaches. The reduced transparency and explainability in design of these systems hinders "safe-by-design" arguments, unless overly restrictive measures are used to constrain system behaviour. This has contributed to a shift towards statistical validation to demonstrate an acceptable level of risk due to residual systematic errors. We assert that arguing both the sufficiency of the specification and the representativeness of statistical validation requires some **model of the system and its environment** that includes probabilistic properties of potential triggering conditions and their causalities. A model-based approach to the assurance of autonomous ML-based systems will increase the **effectiveness of safety assurance arguments** by directly addressing assurance uncertainty caused by environmental and system complexity coupled with the use of opaque data-driven AI approaches. This approach also forms the basis of our ongoing research supported by the ARIA

Safeguarded AI programme[13]. A significant challenge in this approach is ensuring that relevant properties are modelled whilst avoiding exponential increases in model complexity and manual . To address this scaling problem, AI itself has been proposed as a means of supporting the generation and analysis of suitable world models (Dalrymple, et al., 2024). Nevertheless, residual assurance uncertainty will inevitably remain depending on the fidelity of these models. **Reflective assurance** is therefore needed that acknowledges and explicitly uncovers uncertainty within the assurance argument itself. This residual assurance uncertainty must then be systematically reduced to an acceptable level over time via iterative and through-life assurance principles.

Finally, these technical viewpoints must be considered within a holistic framework guided by ethical and legal principles where the deployment and operation of the systems are carefully monitored within an evolving regulatory framework. The approaches laid out in this paper, taken together, may be sufficient to deploy the current generation of automated driving systems in a manner by which the residual risk can be both controlled and accepted. However, as the use cases and technology evolve, and with the advent of end-to-end learning approaches to automated systems control and agentic AI, our safety assurance approaches will once again be severely challenged. Perspectives on AI safety championed by researchers such as Bengio (Bengio, 2017) and LeCun (LeCun, 2022) have emphasised the importance of structured, self-consistent world models that tightly integrate perception, reasoning, and action. We believe that such approaches would provide a far stronger basis for "assurable" AI systems, particularly in open-ended, safety-critical domains, as the underlying world models can be scrutinised and validated as part of the framework outlined in this paper and summarised in **Figure 4**.

Acknowledgments The authors contributions to this paper were supported by the ARIA Safeguarded AI programme.

Bibliography

Åsljung, D., Joans, N., & Jonas Fredriksson. (2017). Using extreme value theory for vehicle level safety validation and implications for autonomous vehicles. *IEEE Transactions on Intelligent Vehicles, 2*(4), 288-297.

Bengio, Y. (2017). The consciousness prior. *arXiv preprint arXiv:1709.08568*.

Burton, S., & Herd, B. (2023). Addressing uncertainty in the safety assurance of machine-learning. *Frontiers in Computer Science, 5*, 1132580.

Burton, S., & Herd, B. (2024). Uncertainty-aware evaluation of quantitative ML safety requirements. *International Conference on Computer Safety, Reliability, and Security*, 391-404. Springer.

Burton, S., & McDermid, J. (2023). *Closing the gaps: Complexity and uncertainty in the safety assurance and regulation of automated driving.*

[13] https://www.aria.org.uk/opportunity-spaces/mathematics-for-safe-ai/safeguarded-ai

Burton, S., Habli, I., Lawton, T., McDermid, J., Morgan, P., & Porter, Z. (2020). Mind the gaps: Assuring the safety of autonomous systems from an engineering, ethical, and legal perspective. *Artificial Intelligence, 279*, 103201.

Burton, S., Hellert, C., Hüger, F., & Mock, M. (2022). Safety Assurance of Machine Learning for Perception Functions. In *Deep Neural Networks and Data for Automated Driving: Robustness, Uncertainty Quantification, and Insights Towards Safety* (p. 335). Springer Nature.

Burton, S., McDermid, J., Garnet, P., & Weaver, R. (2021). Safety, complexity, and automated driving: holistic perspectives on safety assurance. *IEEE Computer, 54*(8), 22-32.

Butz, M., Heinzemann, C., Herrman, M., Oehlerking, J., Rittel, M., Schalm, N., & Ziegenbein, D. (2020). SOCA: domain analysis for highly automated driving systems. *IEEE 23rd international conference on intelligent transportation systems (ITSC)* (pp. 1-6). IEEE.

Dalrymple, D., Skalse, J., Bengio, Y., Russel, S., Tegmark, M., Seshia, S., . . . Ammann, N. (2024). Towards Guaranteed Safe AI: A Framework for Ensuring Robust and Reliable AI Systems. *arXiv preprint: arXiv:2405.06624*.

Erdi, P. (2007). *Complexity Explained.* Springer Science.

Erz, J., Burton, S., & Sax, E. (2025). Extending the Automotive Safety Life Cycle Towards Operational Safety of Automated Driving. *IEEE international automated vehicle validation conference.* IEEE.

Favaro, F., Fraade-Blanar, L., Schnelle, S., Victor, T., Pena, M., Engstrom, J., . . . Smith, D. (2023). Building a credible case for safety: Waymo's approach for the determination of absence of unreasonable risk.

Gansch, R., Putze, L., Koopmann, T., Reich, J., & Neurohr, C. (2025). Causal Bayesian Networks for Data-Driven Safety Analysis of Complex Systems. *International symposium on model-based safety and assessment* (pp. 222-237). Springer.

Hawkins, R., Habli, I., & Kelly, T. (2013). The principles of software safety assurance. *31st International System Safety Conference* (pp. 12-16). Boston, Massachusetts USA: The International System Safety Society.

Hawkins, R., Kelly, T., Knight, J., & Graydon, P. (2011). A New Approach to creating Clear Safety Arguments. *Advances in Systems Safety*, 3-23. London: Springer London.

Hawkins, R., Osborne, M., Parsons, M., Nicholson, M., McDermid, J., & Habli, I. (2022). Guidance on the safety assurance of autonomous systems in complex environments (SACE). *arXiv preprint arXiv:2208.00853*.

Jonasson, J., & Rootzen, H. (2014). Internal validation of near-crashes in naturalistic driving studies: A continuous and multivariate approach. *Accident analysis and prevention, 62*, 102-109.

Josang, A. (2016). *Subjective logic* (Vol. 3). Springer.

Kelly, T. (2014). Software Certification: Where is Confidence Won and Lost? *Addressing systems safety challenges*.

Knight, F. (1921). Risk, Uncertainty and Profit. University of Illinois at Urbana-Champaign's Academy for Entrepreneurial Leadership Historical Research Reference in Entrepreneurship.

LeCun, Y. (2022). A path towards autonomous machine intelligence version 0.9. 2, 2022-06-27. *Open review, 62*(1), 1-62.

Lovell, B. (1995). *A Taxonomy of Types of Uncertainty.* Portland, OR: Portland State University.

Porter, Z., Calinescu, R., Lim, E., Hodge, V., Ryan, P., Burton, S., . . . Molloy, J. (2025). INSYTE: a classification framework for traditional to agentic AI systems. *ACM Transactions on autonomous and adaptive systems, 20*(3), 1-39.

Porter, Z., Habli, I., McDermid, J., & Kaas, M. (2024). A principles-based ethics assurance argument pattern for AI and autonomous systems. *AI and ethics, 4*(2), 593-616.

Roudposhti, S. S., Hawkins, R., Burton, S., Hodge, V., Paterson, C., & Habli, I. (2025). A Case Study on defining traceable Machine Learning Safety Requirements for an Automotive Perception component. *36th IEEE International Symposium on Software Reliability Engineering*.

Sifakis, J., & Harel, D. (2023). Trustworthy autonomous system development. *ACM Transactions on embedded computing systems, 22*(3), 1-24.

Songchitruksa, P., & Tarko, A. (2006). The extreme value theory approach to safety estimation. *Accident analysis and prevention, 38*(4), 811-822.

Varadarajan, S., Bloomfield, R., Rushby, J., Gupta, G., Murugesan, A., Stroud, R., . . . Wong, I. H. (2023). Clarissa: foundations, tools and automation for assurance cases. *IEEE/AIAA 42nd Digital avionics systems conference (DASC)*, 1-10. IEEE.

Walker, W., Harremoës, P., Rotmans, J., Vand Der Sluijs, J. P., Van Asselt, M. B., Janssen, P., & Krayer von Krauss, M. P. (2003). Defining uncertainty: a conceptual basis for uncertainty management in model-based decision support. *Integrated Assessment, 4*(1), 5-17.

Werling, M., Faller, R., Betz, W., & Straub, D. (2025). Safety integrity framework for automated driving. *arXiv preprint arXiv:2503.20544*.

Zou, J., Stefanakos, I., Roudposhti, S. S., Burton, S., Calinescu, R., Clegg, K., & Rivett, R. (2026). Structural causal world models for safety assurance of AI-based autonomy. *41st ACM/SIGAPP Symposium on Applied Computing (SAC'26).* ACM.