SRI International

CSL Technical Report SRI-CSL-2024-02R2 • August 7, 2024

Assurance of AI Systems From a Dependability Perspective

Robin Bloomfield (City, Univ. of London) and John Rushby (SRI)

SRI Project 101425 in support of DARPA ANSR Program.
Distribution Statement “A” (Approved for Public Release, Distribution Unlimited).

Computer Science Laboratory • 333 Ravenswood Ave. • Menlo Park, CA 94025 • (650) 326-6200 • Facsimile: (650) 859-2844

Abstract

We outline the principles of classical assurance for computer-based systems that pose significant risks. We then consider application of these principles to systems that employ Artificial Intelligence (AI) and Machine Learning (ML).

A key element in this "dependability" perspective is a requirement to have near-complete understanding of the behavior of critical components, and this is considered infeasible for AI and ML. Hence the dependability perspective aims to minimize trust in AI and ML elements by using "defense in depth" with a hierarchy of less complex systems, some of which may be highly assured conventionally engineered components, to "guard" them. This may be contrasted with the "trustworthiness" perspective that seeks to apply assurance to the AI and ML elements themselves.

In cyber-physical and many other systems, it is difficult to provide guards that do not depend on AI and ML to perceive their environment (e.g., other vehicles sharing the road with a self-driving car), so both perspectives are needed and there is a continuum or spectrum between them. We focus on architectures toward the dependability end of the continuum and invite others to consider additional points along the spectrum.

For guards that require perception using AI and ML, we examine ways to minimize the trust placed in these elements; they include diversity, defense in depth, explanations, and micro-ODDs. We also examine methods to enforce acceptable behavior, given a model of the world. These include classical cyber-physical calculations and envelopes, and normative rules based on overarching principles, constitutions, ethics, or reputation.

We apply our perspective to autonomous systems, AI systems for specific functions, general-purpose AI such as Large Language Models, and to Artificial General Intelligence (AGI), and we propose current best practice and an agenda for research.

Contents

1 Introduction: Assurance for Traditional Systems **3**
1.1 Traditional Systems and their Assurance 5
1.2 From Assurance to Dependability 7

2 Assurance for Systems Extended with AI and ML **10**
2.1 Runtime Verification 11
2.2 The Challenge of Assuring Perception 13
2.3 Assurance through Diversity and Defense in Depth 16
2.4 Summary of Architectural Choices 19
2.5 Operational Design Domains (ODDs) and Micro ODDs 22

3 Assurance of AI Systems for Specific Functions **23**

4 Assurance for General-Purpose AI **31**

5 Assurance and Alignment for AGI **38**

6 Summary and Conclusion **44**

References **48**

1 Introduction: Assurance for Traditional Systems

Humankind has been concerned about the safety of their constructions ever since they started making them. From the beginning, they noted failures, developed good practices, and specified liabilities and penalties. Nearly 4,000 years ago, the Code of Hammurabi stipulated:

> "If a builder build a house for some one, and does not construct it properly, and the house which he built fall in and kill its owner, then that builder shall be put to death" [69, Section 229].

Hazards to safety depend on what a constructed thing does and how it does it. Buildings, boats, bridges, and mines were among the earliest constructions, and the hazards were that they would fall down, break up, or catch fire, so nascent safety engineering and assurance focused (not always successfully) on ensuring that they were of adequate strength and were based on some understanding of the mechanisms of stress and failure. Active systems such as boats not only needed to be strong, but to possess some form of stability so that they would right themselves rather than tip over in wind gusts, and a means of control so they could steer a desired course. And inherently dangerous constructions such as underground mines would need to include escape routes from collapse or fire. These concerns and methods were refined in the industrial revolution as machines such as high pressure steam engines did new things and introduced new hazards. Systems such as railways introduced the need for active procedures to ensure safe operation, such as signaling protocols to prevent two trains using the same track.

Later, control systems became automated, first by mechanical systems such as governors, then by analog electronic systems such as autopilots, and then by digital computers. Protocols and protection systems also became automated, first with interlocks and then with full automation implemented by digital computers.

Systems with control and procedural mechanisms implemented by computers (so-called cyber-physical systems, CPS) drive the state of the art in safety engineering and assurance today [102]. The central tenet of the approach taken, which we call the *dependability* perspective, is that there must be near-complete understanding of how the given system works, what are its hazards, how these are eliminated or mitigated, and how we can be sure all this is done correctly. The evidence and arguments that justify confidence in the claims documenting this understanding constitute what is called an *assurance case* (see later for details and references).

Systems that use Artificial Intelligence (AI) and Machine Learning (ML) are both an evolution and a step change from their predecessors: although they often automate existing systems and procedures, they work in different ways to what has gone before, and they can also do different things. Because they work in different ways, it is difficult to apply established methods for safety engineering and assurance

to AI and ML, even when they are used in familiar or slightly extended contexts, such as automated control and autonomous systems. In particular, the behavior of systems based on ML is developed experimentally, so their inner workings are opaque and do not support the understanding required for justified confidence.

And because they can be used to do new things, some AI and ML systems introduce entirely new hazards: potentially, they can substitute for people in activities that previously required human levels of perception, language, intelligence, and judgement, and where failure can go beyond physical harm and can affect personal wellbeing, relationships, and society at large. Beyond this fairly incremental progression lies the step to Artificial General Intelligence (AGI), with potentially superhuman performance on significant activities, plus imagination, agency with independent goals, and possibly consciousness.

Our aim in this report is to identify and briefly describe issues, possible methods, and difficulties in assurance for systems with significant AI and ML content. We do this mainly from the dependability perspective, where AI and ML are accommodated by *guarded architectures* that use "guards" or "monitors" to provide assurance for the overall system with few or no assumptions on the AI and ML components.

The dependability perspective may be contrasted with what we call the *trustworthiness* perspective, which does claim some assurance in the behavior of AI and ML components that have been developed, analyzed, tested, augmented, or restricted in various ways.[1] Both perspectives have merit and in practice there is a continuum or spectrum between them. In particular, we envisage guarded architectures that are recursively structured where "first level" guards might use some AI and ML and themselves be guarded by simpler systems, eventually bottoming out on conventionally engineered and assured guards so that the overall architecture provides "defense in depth." The architecture and its assurance will vary according to how much assurance "credit" is taken for trustworthiness or diversity of AI and ML components [26]. We call this the dependability/trustworthiness spectrum; a "pure" dependability perspective takes no credit for trustworthiness of AI and ML components, while a "pure" trustworthiness perspective claims full assurance credit for those components.

In the remainder of this section, we describe the dependability perspective on assurance for traditional systems that do not employ AI or ML. Subsequent sections introduce increasing amounts of AI and ML and we discuss approaches and concerns regarding their assurance from perspectives toward the dependability end

[1] Terminology across different fields is always difficult and sometimes contentious; the field of dependability regards "trustworthiness" as a synonym for "dependability," whereas AI tends to use "safety" as the required property (e.g., the UK Government has an AI Safety Institute) and "trustworthiness" as the means for achieving and assuring it. The adjective "trustworthy" carries a somewhat anthropomorphic tone that we discuss in Section 4 and so we generally prefer the more neutral, engineering terminology of dependability. However, we need a name for the alternative perspective and we use "trustworthy" for that purpose.

of the spectrum. We invite others to provide complementary studies toward its trustworthiness end. We stress that the purpose of assurance as we present it is not to impose a brake or burden on development, but to support innovation by anticipating downstream hazards and suggesting creative ways to mitigate them.

1.1 Traditional Systems and their Assurance

State of the art non-AI cyber-physical systems such as aircraft flight control, safety systems such as nuclear shutdown, and all manner of systems within critical infrastructure, medical devices, personal gadgets and much else are generally engineered and assured for suitably high levels of safety and other required attributes, such as security or effectiveness, all generically referred to as *dependability* [81, 105]. In outline, the process for doing this begins with identification of the potential hazards that the proposed system might entail. A hazard is a circumstance with an unacceptably high risk of leading to harm or other undesired outcome. Hazard analysis is conducted in the context of assumptions about the environment in which the system will operate and is not an exact science: even its most effective methods can be imperfect and their application requires skill, knowledge and experience [80, 119].

As hazards are identified, the system and its evolving design are adjusted to eliminate or mitigate them. For example, if fire is a hazard, we may try to eliminate it by removing sources of ignition and fuel; if that is impossible or inadequate, we can try to mitigate the hazard by adding a fire suppression system. But then we have new hazards concerning failure of that suppression system. Note that we usually try to separate those parts of the system concerned with elimination and mitigation of hazards from those parts that deliver its general functionality: the goal is to minimize the size and complexity of those parts that need the highest levels of assurance. We will also want to protect these critical parts from the rest of the system: a practice known as *partitioning* [144]. Of course, some aspects of the system's general functionality may also be considered critical and they, too, will be partitioned to the extent possible, and subject to assurance. And some auxiliary functions such as logging may also be considered critical as they will be needed to support forensic investigation in the case of failure (consider the difficulty in conclusively demonstrating failures of the British Post Office Horizon system [40, 117]).

After some rounds of iteration on hazard identification and modifications to the system goals and design, we will have a set of *requirements* for the critical *desired behavior* of the computer control system that, with high confidence, ensures dependability of the overall system in its assumed environment. Identification and articulation of properties assumed about the environment are fundamental to formulation of requirements and are often the most difficult and fault-prone aspects of the entire system engineering endeavor. The analysis and reasoning that shows that

the requirements ensure safety and other critical properties within their environment is an assurance task that we term *dependability requirements validation*.

Requirements concern what the system will do, not how it will do it, so they should largely be described in terms of changes the system is to bring about in the environment (this is a key insight due to Michael Jackson [82]). How the system will do its task is developed in *specifications* for the *defined behavior* of the system and the *architecture* of its components. Architecture is a generalization of partitioning (often portrayed by "boxes and arrows" diagrams) and its purposes are to identify *fault containment regions* that limit *fault propagation* among components, to identify *critical components* and limit their complexity (because complexity is a source of faults and also makes it more difficult to discover what faults may be present), and generally organize things so that dependability relies on only the architecture and the defined behavior of the critical components [30].

We then *implement* the system according to its specifications and architecture. The mechanisms that ensure an architecture is faithfully represented in the system implementation are among the most difficult engineering challenges in computer science (involving operating systems, "buses," distributed consensus, state-machine replication, transaction mechanisms etc.) and should employ only well-attested techniques and products with no "homespun" solutions [145]. During implementation, we may discover new hazards and the whole process iterates: the new hazards cause revision to the requirements[2] and their safety validation, and also to the specifications and hence to the implementation.

Assurance is developed during and following this process. After **dependability requirements validation**, assurance divides into three **verification** tasks. (Verification differs from validation in that, in principle, it can be performed with perfect accuracy.)

Intent. The specifications must be shown to be correct and complete with respect to the requirements, subject to properties of the architecture and assumptions about the environment

Correctness. The implementation must be shown to be correct and complete with respect to the specifications, subject to properties of the architecture and assumptions about the environment.

Innocuity. Any part of the implementation that is not derived from the requirements must be shown to have no unacceptable impact.[3]

[2] Confusingly, these revisions are often called *derived requirements* (the term comes from avionics); it is confusing because their essence is that these requirements were *not* derived during the main process of requirements development.

[3] Software libraries provide an example of implementation content that is not derived from requirements: we might require only the trigonometric functions, but the whole library is installed as part of the implementation.

Different industries have their own standards and guidelines that codify aspects of this process, often in great detail; the very generic and abstract description given above is based on the *Overarching Properties* (OPs) proposed as the basis for future civil aircraft certification in the USA [75].[4]

Each of the assurance validation and verification tasks states that some properties "must be shown" to hold; by this, we mean that there must be reasons why the properties hold, and these reasons must be clearly articulated and justified. The state of the art for doing this is an *assurance case* (a generalization of safety cases [1,91]) that provides an organized presentation based on *claims*, *evidence*, and *argument* [24, 147]. Claims identify properties of the system and/or its environment; evidence refers to observations, measurements, or experiments on the system or its means of construction or on its environment that justify certain claims; and the argument uses the evidence to establish a hierarchy of claims culminating in a significant *top claim*. The arguments of an assurance case are not free form but *structured* as hierarchy of argument steps, each of which establishes a "parent" claim on the basis of one or more "child" claims (we usually say *subclaims*) established at lower levels, or by evidence. A portion of a graphical rendering of an assurance argument is displayed in Figure 1; our preferred treatment of modern assurance cases, which we call Assurance 2.0, is presented elsewhere [28] and builds on the ideas that a strong assurance case should be *indefeasible* [29, 148], based on established *theories* [171], and subjected to *dialectical examination* [27].

1.2 From Assurance to Dependability

The focus on dependability validation and verification with overall justification presented as an assurance case might seem like good practice and a sensible way to develop high quality systems, but why is it needed for assurance? Why don't we just test the thing? Indeed, whenever there is a major systems failure, the first reaction of the press and public is "they didn't test it enough." But in fact, testing is insufficient and the reason is the extraordinary levels of confidence required for safety and other critical properties and, consequently, the infeasibly large number of tests that would be required to validate them by observation alone. We give a few numbers for illustration.

In commercial airplanes, "catastrophic failure conditions" (those "which would prevent continued safe flight and landing") must be "so unlikely that they are not anticipated to occur during the entire operational life of all airplanes of one type" [56]. The "entire operational life of all airplanes of one type" is about 10^8 to 10^9

[4] OP concerns software assurance, so dependability/safety validation is outside (occurs prior to) its scope. Also, OP speaks of desired and defined behavior rather than requirements and specifications.

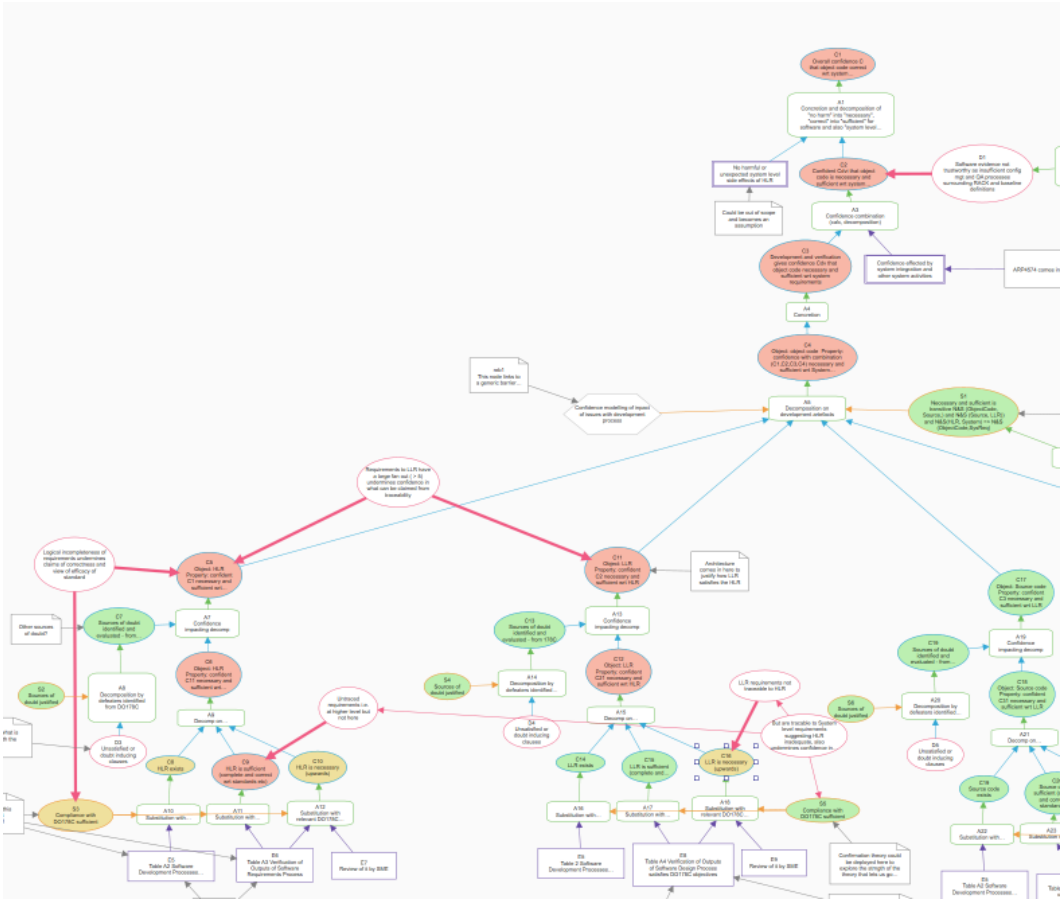


Figure 1: Portion of Graphical Rendering of an Assurance Case Argument

flights for modern airplanes. With an average flight duration of about 90 minutes, this requires a critical failure rate no worse than about 10^{-9} per hour.[5]

Cars are among the most dangerous consumer goods with about 40,000 deaths per year in the United States and a fatal accident rate of a little over 10 per billion miles. It is intended that self-driving cars should be safer than human drivers, so it might seem reasonable (even though driving error is not the only cause of accidents) to set the target at no more than 1 death per billion miles, which is around 10^{-7} per hour, given an average speed of 30 mph. However, that is a rather technical assessment that pays no attention to likely public reaction. Herbert Diess, the former CEO of Volkswagen is quoted on their website with a more realistic assessment: "A ratio of ten-to-one is nowhere near good enough. We have approximately 3,200

[5]There are many measures for reliability, such as failure rate, probability of failure on demand, probability of fatality per mile, etc. And the underlying system models may use discrete or continuous time, with Bernoulli or Poisson failure processes, etc. The general conclusions drawn here are robust to all these choices and so we do not describe them in detail.

traffic fatalities in Germany each year. It would be a disaster if we had even 320 deaths due to driverless cars." Thus, it is widely accepted that the safety target for self-driving cars should be 100 or even 1,000 times better than human drivers,[6] which brings us into, or even beyond, the requirements for commercial aircraft. But commercial aircraft do get assured and certified and their safety record justifies this, so why is 10^{-9} seen as such a challenge?

The answer is that we do not assure aircraft solely by testing. If we test a system for n hours and observe no failures, then in the absence of other information, the best prediction we can make is that the likelihood of no failures in another n hours is about 50–50 [113, p. 73, and sidebar on p. 74]. Hence, to secure assurance for failure rates of 10^{-9} per hour we would need to test the system for about 10^9 hours, or around 115,000 years. Even with 1,000 copies of the system on test, this is still well over 100 years of continuous operation and is completely infeasible [37, 90, 113].

The purpose of the classical methods of assurance described earlier is to give justified confidence that the system contains no (dependability) faults and hence will have no dependability failures. Confidence can be expressed as a subjective probability, so if we are 95% confident that traditional assurance works, that means we estimate there is only 5% chance that the assured system contains faults. We can now use testing to explore the existence of those potential faults but, unlike the previous case, we know something about the system so when we see n hours with no failures, we can conclude (by what is called Conservative Bayesian Inference, CBI) that we are likely to see another $10n$ with no failures [163, 182]. This reduces the amount of testing required and another idea reduces it still further. This is Bootstrapping [22]: we need assurance for 10^9 hours, but this is over the lifetime of the system. When the system is first deployed, we might be satisfied to know there will be no dependability failures in the first year, and we will have only a few instances of the system operational in that time. So we might need confidence for only, say, 10^5 hours, and testing for this, given prior assurance and CBI, requires only 10^4 hours, which is perfectly feasible. After the first year, we might seek confidence for the next year and for the larger number of systems now installed, but we will have the operational experience of the first year and that should be sufficient (given the tenfold multiplier of CBI) to deliver the required confidence in safe operation, and so on for subsequent years.

But suppose experience in operation does reveal a failure, hopefully not catastrophic (in commercial aircraft, for example, it is required that no catastrophic failure may be caused by a single fault). Any such failure reveals an unanticipated fault, arrival rate, circumstance, or hazard, and these could be precursors

[6] This sensitivity is illustrated by a recent accident in San Francisco: a car with a human driver hit a pedestrian and threw them under the wheels of a Cruise self-driving taxi, which then dragged the victim for several yards. Subsequently, Cruise lost their license to operate in San Francisco (although their poor initial response to the incident probably contributed to this harsh reaction) [97].

to a catastrophic failure. Consequently, commercial airplanes operate in a legal and ethical framework where all incidents and accidents are promptly reported and dispassionately investigated. The FAA issues Airworthiness Directives mandating workarounds or corrections to detected faults; in extreme cases it may temporarily ground the fleet (as it did for the 737 MAX in 2019–2020). Bishop [21] constructs a statistical model for this scenario and shows that, under plausible assumptions, detection and repair of faults significantly increases long run safety, even if the fleet continues to operate after a fault has been discovered, and even if repairs may be imperfect. It follows that there is much value in monitoring, analyzing and, if warranted, correcting all non-trivial failures and their precursors.

We can now see that traditional approaches to assurance, such as described for OP, give us justifiably strong confidence that the system satisfies its critical properties; this can then be augmented by testing and operational experience to deliver (via CBI and Bootstrapping) justified confidence that the system satisfies its dependability goals, and this can be reinforced by monitoring during operation. In the following sections, we consider systems with AI and ML components that do not lend themselves to traditional methods of assurance and may therefore be ineligible for CBI, and we explore how, and to what extent, dependability can nonetheless be achieved and guaranteed.

2 Assurance for Systems Extended with AI and ML

In this section, we focus on systems that do fairly traditional things but are now extended with capabilities enabled by AI and ML. Autonomous CPS such as self-driving cars are canonical examples.

For traditional assurance, there must be good reasons why we believe the system achieves its dependability goals and those reasons are documented and justified in its assurance case. Systems that use AI and ML pose challenges to this approach because, rather than performing actions that are effective and safe for reasons that can be articulated and verified, a system that uses ML operates by learning suitable behavior during a period of training. Training typically defines empirically effective "weights" in a *deep neural network* (DNN); there will often be millions, or even billions, of individually adjusted weights.[7] The hope is that if the system works correctly on the training examples, then it will work correctly on all similar examples.

An alternative to ML is *symbolic* AI, which uses automated deduction (theorem proving) to derive conclusions from premises composed of a set of axioms describing

[7] There are other ML techniques, such as Support Vector Machines [161] but, despite different mechanisms, these pose similar challenges to assurance as do neural networks. Similarly, the specific ML architecture in which neural networks are employed, such as reinforcement and inverse reinforcement learning, or large generative language and diffusion models, has little impact on the fundamental difficulty of assuring ML.

some aspect of the world plus observations about the current state of the world. It is possible to guarantee validity of some methods of automated deduction (e.g., SAT and SMT solvers with certificates [120]), but soundness depends on the choice of premises, which may be unvalidated and derived from an informal or empirical model, and also on the computational resources available (deduction generally requires exponential time, or worse, so we may need to accept whatever partial analysis can be accomplished in fixed time). "Expert systems" were a type of symbolic AI popular in the 1980s where deductive procedures were applied to a collection of "rules" that axiomatized some domain. The concern, and one of the reasons for the demise of these systems, was that individually reasonable rules could collectively be inconsistent or incomplete, resulting in faulty output [143]. Modern symbolic AI uses improved technology but its challenge to assurance remains largely unchanged. A combination of symbolic AI with ML is a popular current approach known as *neurosymbolic* AI. The strengths and weaknesses of the two approaches seem to be complementary, but this does not assuage their assurance problems.

Traditional assurance requires good understanding of how the system works because tasks such as intent verification must show that certain properties hold in *all* circumstances. This is infeasible for most AI and ML components because we lack detailed understanding of their operation,[8] but an alternative or constituent part of the overall system assurance process can be to check that properties hold in the circumstances actually encountered during operation. This is *runtime verification* [142] or, more boldly *runtime certification* [146], where components, often generically referred to as *guards* (or *monitors* [71, 121]), are added to the system to check its behavior against its required or specified properties, or conservative simplifications of these. If a check fails, then the system must take some remedial action to maintain or restore safety. Both checking and remediation add complexity to the system and may themselves introduce failures and hazards, so this approach requires careful engineering [100]. Nonetheless, a plausible approach is to guard AI and ML elements with conventionally engineered components that perform runtime verification and can be assured in the conventional way. This approach is endorsed in some industry guidelines such as F3269-17 for unmanned aircraft [7].

2.1 Runtime Verification

To investigate runtime verification for systems with AI and ML components, we need some general understanding of the likely overall system architecture and the properties that will be checked. The top-level structure of almost any system that employs AI or ML follows from a single insight, which is that any entity that interacts effectively with some aspect of the world must have a *model* of that aspect of the

[8] Recently, there has been progress in associating learned "concepts" with specific clusters of artificial neurons [166], but this is still some way from the understanding required for assurance.

world [45]. In particular, cyber-physical systems are always based on a model of the controlled "plant" and its environment [59]. For pure control systems, the model may be a collection of differential equations; for systems that involve procedures it may add state machines; and for full CPS it will include integrated formalisms such as timed and hybrid automata.[9]

The model may be used in development but be represented only indirectly in the final system (e.g., as control laws in classical control engineering), or it may be partially defined at design time and built in to the system with parameters that can be adjusted at runtime (as in adaptive control systems), or it may largely be constructed at runtime and represented explicitly within the system (as in model-predictive control and autonomous systems). A characteristic of most systems that employ AI and ML is that these capabilities are used in building and maintaining a model of the "world" (i.e., its environment, such as the locations of other road users) at runtime; this function is generally referred to as the *perception* (sub)system. Optionally, AI and ML may also be employed in the *action* (sub)system, which uses the world model to calculate and execute behavior that will advance the system's progress toward its goal, while maintaining dependability.[10]

It is often possible to guard AI or ML-generated actions with highly assured conventional software that checks their safety against the world model. However, if the world model is constructed by a perception system that uses AI or ML then we have to ask how its own accuracy can be assured or guarded: it does no good to run safety checks against a faulty model. One possibility is for the guard to use a different model that can be assured.

We can therefore distinguish two classes of guarded AI-enabled systems according to the nature of their runtime verification. *Assuredly guarded* systems are those whose safety and other critical properties can be checked and enforced by assured guards that do not themselves use AI or ML, neither for perception nor action. In cases where the guards are so strongly assured that they can be considered "possibly perfect" (or "probably fault-free"), it is possible to make very strong claims for dependability of the overall system [112,183]. The second class is *unassuredly* guarded systems, where the guards themselves depend on AI or ML, typically in their per-

[9] A model can be thought of as a simulation (or "digital twin") of the plant and its environment; "model based system/software engineering" (MBSE) builds and experiments with explicit simulations of these plus the intended system using tools such as StateFlow/Simulink and derives the requirements and specification, and sometimes the implementation (and sometimes does that automatically), from the system simulation. A criticism of this approach is that the focus on simulation introduces implementation concerns too early into the process (it is difficult to simulate requirements stated as constraints unless theorem proving is used, so specifications are often substituted for requirements) and thereby compromises dependability requirements validation.

[10] There is often substructure within the action system: generally a *decision* system that plans what to do (e.g., "move into the nearside lane between cars `a` and `b`") and an *execution* system that manipulates the controls (accelerator, brakes, steering, horn) to perform that plan. These details are unimportant at our level of discussion.

ception systems. See, for example, the architectures listed on page 20, where those that include a traditionally assured backup guard are assuredly guarded and others (except number 1, which is not guarded) are unassuredly guarded.

An example of the first class is an unmanned aircraft controlled by AI and ML that is deemed safe as long as it remains within some specified portion of airspace (a "geofence"). This is a constraint that can be checked by a conventional navigation system and enforced using a conventional guidance system as an override when violations are detected [48]. Many robots can be guarded by "virtual cages" of this kind. Notice also that when formulating the initial "concept of operation" for a system, it may be possible to adjust the concept so that assurable guards become feasible. For example, rather than an autonomous shuttle bus sharing its route with other vehicles and pedestrians, it could use a dedicated and isolated track, and would not then require a sophisticated AI-based perception system.

An example of the second class is a self-driving car where the action guard checks AI-generated actions for safety and other dependability properties against a model of its environment—i.e., location of other road users and pedestrians, interpretation of traffic signs and markings, detection of road layout, etc. To verify the guard we need assurance for accuracy of its model, and to do this in the traditional way we would want to see its requirements (i.e., what it should represent and what it means for it to do so with acceptable accuracy), its hazards (e.g., inaccuracies that can lead to harm), together with specifications and an implementation that can be shown to achieve the requirements and mitigate the hazards. Typically, none of this is present for models constructed by an AI perception system that uses ML to interpret its sensors, although recent developments are starting to explore some of these topics [25, 39, 76].

2.2 The Challenge of Assuring Perception

We have seen that an action subsystem can generally be assuredly guarded, but is dependent on a model of the world constructed by a perception subsystem that is harder to guard.

A possible compromise is for the action guard to use a simpler, conservative model constructed by conventional and highly assured software. This is sometimes seen in self-driving cars where the guard is a simple system for automated collision detection and emergency braking, similar to the Automated Driver Assistance System (ADAS) that is provided for some human-driven cars.[11] However, this does not provide full assurance because forward collisions are not the only hazards (Mehmed

[11] The US National Highway Traffic Safety Administration recently finalized a rule (FMVSS No. 127) that will require Automatic Emergency Braking (AEB), to be standard on all passenger cars and light trucks by September 2029. Tests by the American Automobile Association found that current AEBs (which are built to a lower, voluntary standard) are somewhat effective against rear-end collisions, but not at all effective against sideways collisions at intersections.

and colleagues cite data from a NHTSA study of 5.9 million human-driver accidents classified in 37 categories [122]), nor are human accidents necessarily good models for failures of an autonomous system, and nor is emergency braking attractive as the sole method of hazard mitigation. And notice that to prevent excessive activation of the emergency system, the primary system of a self-driving car must take its behavior and capabilities into account. For example, the primary action subsystem may calculate (correctly) that a certain maneuver is optimal and safe, but that it is also likely to activate the more hair-trigger response of the emergency system, and so it must choose a different behavior. In general, systems employing defense in depth must be designed and developed as a whole to ensure that unintended differences among the layers do not cause unnecessary loss of availability, while safety arguments based on diversity are nevertheless preserved. These are challenging requirements.

Furthermore, accidents and collisions are not the only hazards that should be considered: for example, the City of San Francisco has reported dozens of incidents where robotaxis interfered with emergency responders, and there must be many other circumstances where self-driving cars increase risk or inconvenience for others without themselves being involved in a collision.

For verified dependability, assurable guards must detect all hazards and mitigate them safely, without excessive false alarms. It is possible that a more comprehensive suite of verifiable ADAS-like guard functions could do better, but they would have to steer a difficult path among incomplete coverage, false alarms, and unattractive, abrupt, responses. A contrary point of view is that although there may be many circumstances leading to accidents, the exact circumstances are irrelevant to "last second" detection as there are only a few possible emergency responses: essentially, braking or evasive action, and so an ADAS-like guard (or a suite of such guards) could be an acceptable means of assurance, provided we can develop assurance that it will always select and perform appropriate emergency responses (see Section 2.5).

An argument against simple ADAS-like guards is that more sophisticated perception could detect hazardous situations earlier and provide less abrupt mitigation. Accordingly, it is worth considering guards that do use AI and ML and asking whether they can be assured for trustworthiness within an overall approach that remains close to the dependability end of the dependability/trustworthiness spectrum. For example, we could imagine an "ML-friendly" adjustment to traditional assurance where we do construct requirements and identify hazards, and then develop training data of "sufficient" size, coverage, and quality[12] to encompass all of these and use it with a well-regarded ML toolset to generate the perception capabilities desired. In fact, this is exactly what is proposed by several groups engaged in research and development of autonomous systems [4,19,49], although they generally

[12]ML requires such a vast quantity of training data that the training sets are often labeled automatically, by another AI system, and therefore cannot be considered high quality.

consider primary systems rather than guards, and only one focuses specifically on perception [154].

The hazard most generally recognized in perception using ML is lack of "robustness," meaning that small changes in sensor data may cause its ML-generated interpretation to change abruptly. This concern is validated by so-called "adversarial examples" [165]. Typically demonstrated on image classifiers, the examples are deliberately constructed minor modifications to an input image that are indiscernible to a human observer but cause an ML classifier to change its output, often drastically and inappropriately. Image masks have been developed that will cause misclassification when overlayed on any image input to a given classifier, and there are universal examples that will disrupt any classifier [124]. Furthermore, there are patterns that can be applied to real-world artifacts (e.g., small images that can be stuck to traffic signs) that will cause them to be misread by an image classifier [35].

There is much work on detection and defense against adversarial attacks (see [77] for a survey) and on the threats posed by general lack of robustness [104, 172]. A problem with all this work is that the techniques for guaranteed robustness have so far scaled only to relatively simple systems and not, for example, to the object detector of a self-driving car. And, more importantly, robustness is not the topic we really care about: we want assurance of accurate perception. There is work that verifies contracts on some aspects of perception, notably detection of traffic lanes [8], but this particular problem can also be solved without ML [50] (although those solutions may also be hard to verify).

Accurate perception for world models in general depends not only on robust interpretation of individual sensors but also on effective sensor fusion. This is illustrated by the fatal accident between an Uber self-driving car and a pedestrian walking a bicycle in Arizona on 18th March 2018 [125]. The Uber car used three sensor systems (cameras, radars, and lidar) and fused them using a priority scheme that delivered a "flickering" identification of the victim as the sensor systems' own classifiers changed their identifications, and as fusion preferred first one sensor system, then another, as listed below [125, Table 1].

- 5.6 seconds before impact, victim classified as *vehicle*, by radar
- 5.2 seconds before impact, victim classified as *other*, by lidar
- 4.2 seconds before impact, victim classified as *vehicle*, by lidar
- Between 3.8 and 2.7 seconds before impact,
  classification alternated between *vehicle* and *other*, by lidar
- 2.6 seconds before impact, victim classified as *bicycle*, by lidar
- 1.5 seconds before impact, victim classified as *unknown*, by lidar
- 1.2 seconds before impact, victim classified as *bicycle*, by lidar

Consequently, the object tracker never established a trajectory for the victim and the vehicle collided with her even though she had been detected in some form or other for several seconds. The car in this incident used a particularly poor method of sensor fusion and better methods are now employed, but it is not known how to provide assurance for their behavior.

We conclude that approaches based on trustworthy perception do not yet provide an adequate basis for assured world models, although they do represent good practice and can be welcomed and used on that basis.

We are therefore torn between guards that do not use AI or ML and are potentially assurable, but whose interventions are crude and possibly unacceptable, and those that may be acceptable but are unassurable because they use AI and ML for perception. However, there is one last approach that might provide hope: this is an argument based on diversity and defense in depth.

2.3 Assurance through Diversity and Defense in Depth

In our discussion so far, we have been using guards for runtime verification, where overall assurance depends on that of the guard. But we could also argue that any guard, even if it is not assured, will provide redundancy, and its development and implementation can be completely independent and "diverse" from the primary system. Hence, it is plausible that failures of the guard will be independent of those of the primary and the combination could deliver a multiplicative improvement in dependability (i.e., naively, two systems with $pfd \leq 10^{-4}$ give us 10^{-8} overall). We can also imagine a more integrated system where, rather than a primary system and a guard, we have redundant, diverse perception systems with different architectures and training sets contributing to a single consensus model, or to one, but still fused, model for operation and another more conservative one for runtime verification of actions.

The topic of assurance through diversity is large and somewhat contentious. There is little doubt that architectures employing diverse components are generally more reliable than single threads. In particular, there is evidence that "portfolio" or "ensemble" perception systems are more reliable than their individual constituents [87]. The difficulty is in demonstrating that diversity provides benefit in any *particular* case, and in estimating *how much* benefit it provides [111]. In particular, there is no feasible way to validate failure independence (it is a variant on the infeasibility of assurance by testing),[13] nor strong reasons for believing it. This is because some circumstances are just plain hard to interpret and it is possible that all components may then fail together: consider the scenario with the Cruise self-driving taxi described in Footnote 6—would any training set have included pedestrians being thrown under the wheels? Furthermore, these difficult

[13] However, it is feasible to validate modest degrees of independence, and indications of its absence.

cases do not "thin out" as more are considered: the distributions seem to have "fat tails" [98]. Thus, the "multiplicative" argument for assurance by diversity is indeed naive.

However, although diversity alone cannot provide strong assurance, it can provide a useful step in a "ladder" of assurance. Nuclear power generation usually employs such a ladder of protection systems providing *defense in depth*: there is the operational control system, designed to manage the plant efficiently and safely, then a (safety) limitation system that can intervene to ensure the plant behavior stays operational but within some safe envelope, and finally a shutdown system that functions as an assured guard that guarantees to initiate a safe shutdown when safety parameters are violated.[14] The operational and limitation systems are carefully engineered but do not guarantee the dependability goals established for nuclear power: that is accomplished by the assured shutdown system. But the operational and limitation systems are diverse in function and construction and although this does not provide strong assurance, the presence of the limitation system almost certainly reduces demands on the shutdown system. This is beneficial because an emergency shutdown is disruptive and expensive.

The control, limitation, and shutdown systems in this architecture for nuclear power generation all use traditional software, but a similar approach could be used in systems with AI and ML. For example, in a self-driving car the full functionality could be provided by a primary system employing AI and ML that is supported by a diverse system, also employing AI and ML, that is focused on safety, with assurance provided by traditionally engineered ADAS-like emergency backup functions. The diverse primary and safety systems deliver acceptable behavior and reduce demands on the assured backup so that its interventions, though crude, are rare and tolerable. A safety system may introduce false alarms of its own, although these can be verifiably avoided in some circumstances [123]. Example AI safety functions include the "AI Safety Force Field" [126] that avoids creation of unsafe situations, and "REDriver" [164], which monitors proposed trajectories of self-driving cars against Chinese traffic laws.

A criticism of these particular proposals for defense in depth is that the diverse safety-focused system is concerned solely with actions and relies on the same world model as the primary system. If the perception system fails to detect a pedestrian, for example, then it will be absent from the common model and the safety-focused system can deliver no protection: everything will depend on the emergency backup.

One possible mitigation for this hazard is to provide the safety-focused system with separate sensors and perception; alternatively, it could use the same sensors as the primary system but with diverse perception software. Either arrangement

[14] A complete shutdown can be a complex operation involving both automated initiation and intervention by expert operators, depending on the state of the plant and reasons underlying the need to shutdown (cf. the Fukushima incident).

would provide the primary and safety-focused systems with different models of the world (e.g., [70]) but it is debatable whether diverse models are manageable without false alarms and whether the ML-generated model for the safety system could be any more assurable than that for the primary.[15]

It seems preferable to fuse the products of the diverse perception systems into a single model, but, as illustrated by the Uber crash described earlier, that can introduce flaws of its own. A more attractive approach is to construct a single world model using diverse perception systems in a principled way. Conventional perception systems work "bottom up": one or more deep neural nets take sensor data (e.g., images from cameras or point clouds from lidars) and deliver interpretations (e.g., lists of detected objects) that are further processed and fused to produce the world model. One argument against this approach is that it works "backwards" from effect (image) to cause (objects), which is inherently difficult. Another is that it prioritizes fleeting sensor data above the world model, which is the repository of much accumulated information. An alternative approach, and the way human perception is believed to work [18], reasons "forwards" using the model to *predict* sensor data (or basic interpretations thereof) and then applies a form of Bayesian inference known as *Variational Bayes* [93] to optimize the model in a way that minimizes prediction error. Notice that prediction errors provide continuous feedback on accuracy of the world model.

In humans, this mechanism for perception is known as *Predictive Processing* (PP) [176] and it is believed to be coupled with a *dual-process* architecture [60]. The lower-level process, known as "System 1" [89], performs rapid unconscious perception using PP so long as prediction errors are fairly small, indicating the world is evolving as expected. A large prediction error is called a "surprise" and the higher-level "System 2" process intervenes to resolve it using more deliberative cognition. For example, a self-driving car using a perception system of this kind may find that a vehicle some distance ahead that has been seen and correctly predicted for some time disappears, provoking a surprise; System 2 may hypothesize that the disappearance is due to it being occluded by another vehicle that has not been detected; System 2 can then add the hypothesized occluding vehicle to the model and it will be included in future predictions, thereby sensitizing basic sensor interpretation to look for it, and it can also initiate cautious defensive actions such as a lane change.[16]

[15]The model for the safety system could be simpler than that for the primary system. For example, the primary system in a self-driving car must not only detect objects, but their type (car, bicycle, pedestrian etc.) and fine details such as where drivers and pedestrians are looking, because the action system needs to project their likely motion and future position, whereas the safety system could just use objects and surround them with a conservative "safety box." Intuitively, we might think that the simpler model of the safety system could be more trustworthy, but it would be difficult to provide credible assurance for this using current methods.

[16]This scenario is motivated by a Tesla crash in Taiwan on 1 June 2020: https://www.youtube.com/watch?v=ZmHBA_vV39w.

System 2 in this dual-process architecture is a location for diverse refinement of the world model and also, optionally, the guarding of actions [85, 153]. Furthermore, minimization of prediction error provides a principled way to fuse diverse lower-level sensor and perception functions.

A PP perception pipeline can suffer systematic errors that cause it to misperceive some scenes. For a crude example, it could be blind to white trucks seen against clouds: sensors will report no such white trucks, leading to construction of a world model without white trucks; this will generate predictions lacking white trucks that are confirmed by the sensors. So all seems well—until we collide with a white truck. Suitable testing should minimize these circumstances [85], but diversity among sensors and their basic interpretation functions could also reduce them significantly.

A dual-process architecture with System 1 using PP to integrate diverse perception systems and with System 2 providing diverse perception refinement and action guards is an attractive arrangement for autonomous systems. Although assurance for its AI and ML components will be lacking, its overall architecture has a rational structure that can justify modest assurance based on redundancy and diversity. Modest assurance is insufficient for system dependability, but that is not the claim it needs to support: instead, contributing to defense in depth, it supports the claim that it reduces demands on the traditionally engineered emergency backup to a tolerable level,[17] and dependability is assured by the backup.

We have discussed several options for assurance of systems that use AI and ML for autonomy or other advanced functions, so in the following section we provide an enumeration and summary of some of the architectures considered.

2.4 Summary of Architectural Choices

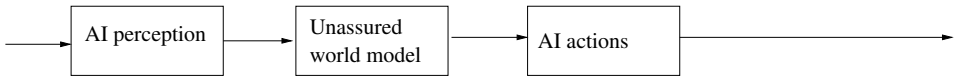
We are concerned with assurance of cyber-physical and similar systems that use AI and ML. We recognized a spectrum of approaches ranging from those that aim to develop trustworthy AI and ML components to those that place no trust in those elements and instead guard them with components whose dependability is achieved and assured by conventional methods.

We divide the critical AI and ML components into a perception subsystem that builds a model of the local world and an action subsystem that uses the model to plan and execute safe and effective actions.

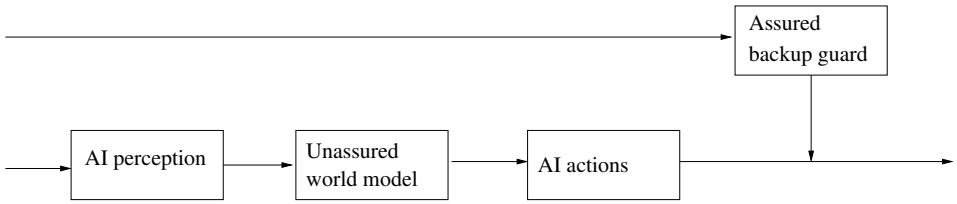
[17] For example, the primary system of a self-driving car might see an object ahead and classify it as a cardboard box and be prepared to drive over it; as it comes closer, the emergency backup will see it as an unclassified obstruction and slam the brakes on; but the safety system will also have seen it as an unclassified obstruction and could have changed lanes to avoid it "just in case." On a larger scale, the safety system might veto a time-optimal route selection by the primary system because it considers it unduly hazardous (e.g., icy).

Before proceeding, we explain the intended interpretation of the diagrams that follow. An arrow from one box to another indicates flow of data, whereas an arrow from a box to a line (see examples below) indicates the ability to intervene or override the flow of data on that line. The incoming arrows on the left indicate data sensed from the environment, while the outgoing arrows on the right indicate control data sent to the actuators.

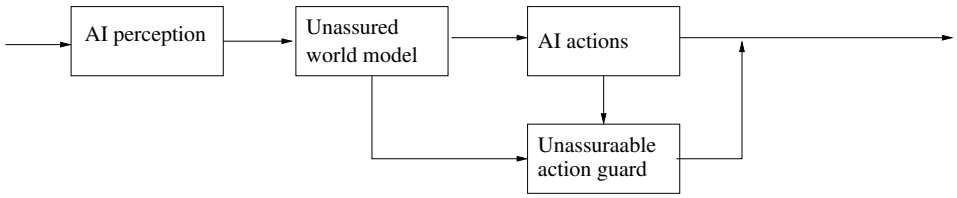
1. We begin at a point on the assurance spectrum where the AI and ML are considered "trustworthy." This does not amount to credible assurance from the dependability perspective so both the world model the action subsystem are considered unassured.[18] Hence, we next focus on the dependability end of the spectrum.



2. As above with the addition of a conventionally engineered and assured guard or backup. The overall architecture is assurable and is viable for applications such as a geofenced vehicle. However, the coverage of the guard and the effectiveness and acceptability of its interventions need careful justification for applications such as a self-driving car where only crude interpretations of the world can be constructed without ML.

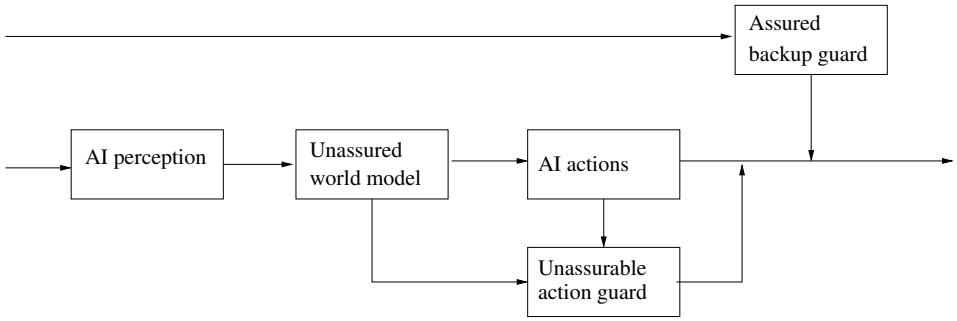


3. As 1 but with a conventionally engineered and assurable guard for the action subsystem, driven from the same model as the primary system. This is vulnerable to errors in perception leading to a flawed world model and so the guard may be verified but it is not assurable and neither is the overall system.

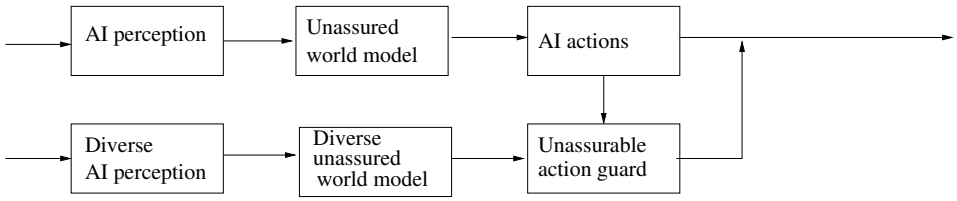


[18] Those who wish to grant some confidence in the trustworthiness of well-examined AI and ML components may read our "unassured/unassurable" labeling as "somewhat assured." "Assured" would then be read as "strongly assured," with "weakly assured" as an intermediate level.

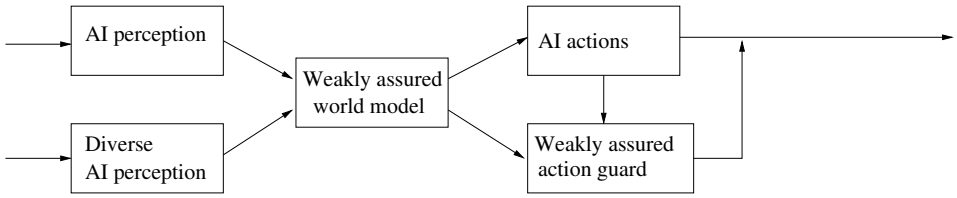
4. Combination of 2 and 3. Overall, this is assurable due to the backup guard. The action guard is not assurable as it is driven by the unassured world model. However, it should reduce demands on the backup, thereby improving on 2.



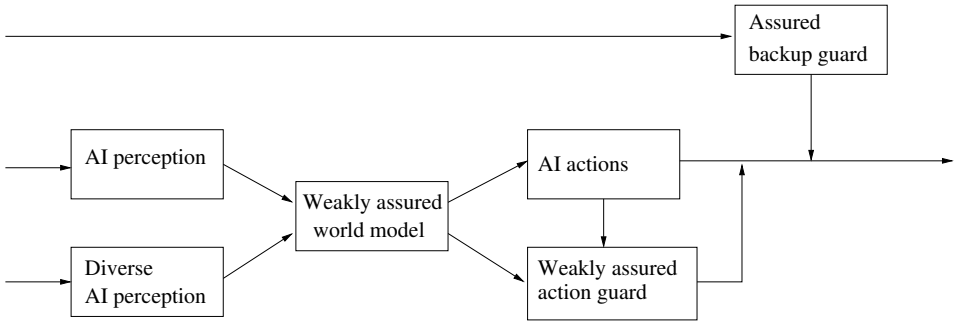
5. As 3 but with diverse perception subsystems driving separate world models for the primary action subsystem and its guard. Diverse perception systems add little without cross-comparison (it could be the guard's model that is flawed), so the guard is unassurable and the overall architecture reduces to 3 and is not assurable.



6. As 3 but with diverse perception systems contributing to a single world model. This is weakly assurable overall: the traditionally engineered guard provides assurance on actions but depends on a world model that is only weakly assured by a diversity argument on perception.



7. Combination of 2 and 6. This architecture is assured by the assured backup guard and improves on 4 because the architecture of 6 provides some assurance for reduction of demands on the backup.



Having considered a range of architectures, we now consider the range of environments in which they may be required to operate.

2.5 Operational Design Domains and Micro ODDs

Assurance goals for a system can be lessened by limiting the range or complexity of circumstances (i.e., environments) in which it is required to operate. For example, self-driving is easier on freeways or in traffic jams than on city streets. These different circumstances are referred to as *Operational Design Domains* (ODDs) and a system may be assured only for specific ODDs and be required to disengage when outside those permitted (alternatively, the system may have different modes of operation in different ODDs). Clearly, the perception system must be augmented to determine when it is in permitted or specific ODDs.

The top level of assurance, namely dependability requirements validation, is strongly focused on hazards and these are largely determined by the chosen ODD. Hence, some approaches argue that assurance should be based on scenarios (i.e., ODDs) rather than technology [42, 92], and others are very focused on identifying hazards associated with chosen ODDs [57]. We propose a variant on these approaches.

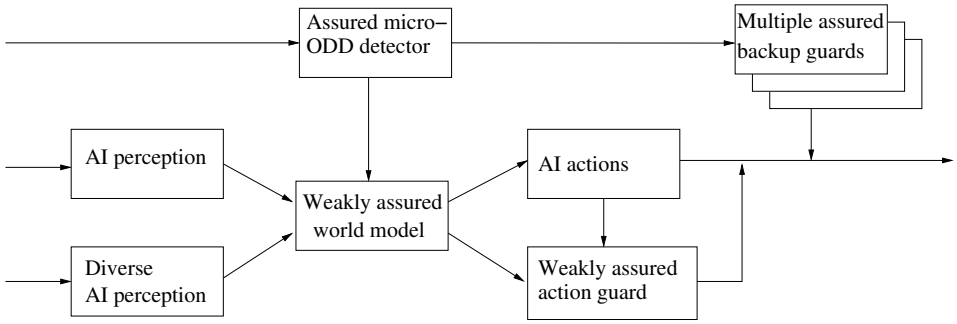
The most credible architectures for assured dependability are those that employ a traditionally engineered and highly assured backup guard as in architectures (2), (4), and (7) of the previous section. However, an argument against these is that interventions by the guard may be too frequent (some due to late detection, others to false alarms) and too crude (e.g., emergency braking). This can be improved by defense in depth as in architectures (4) and (7), where a safety guard that uses AI perception (and is therefore weakly assured at best) reduces demands on the backup.

A further enhancement might be to use multiple backup guards, each specialized to a particular circumstance or ODD. The idea is that forward-looking radar coupled to emergency braking may provide an assured backup arrangement suitable for driving in traffic, but for highway driving it would be better to look further ahead using cameras (having non-AI perception) with speed control as the intervention.

These ODDs would be tailored to assured detection and intervention strategies and might not correspond to traditional ODDs: following [99] we call them micro-ODDs (also written as μODDs). The idea is that for any particular micro-ODD, using the "right" backup guard will deliver superior safety, with fewer false alarms and less disruptive interventions.

This suggests an architecture such as portrayed in (8) below where a portfolio of assured backup guards is coordinated by a detector that recognizes the current micro-ODD and selects the appropriate assured backup guard. Of course, the perception system of the detector must be assured, but its task seems to be rather simple and it is conceivable that it can be performed without AI, or by AI and ML of credible trustworthiness (see [115] for relevant work).

8. Architecture (7) with a portfolio of assured backup guards coordinated by an assured detector that recognizes their matching micro-ODDs.



3 Assurance of AI Systems for Specific Functions

In this section we consider novel non-CPS systems and applications that are made possible or are performed in new ways by the capabilities of AI and ML. These include systems that play games of skill or strategy, those that design things or perform scientific predictions such as protein folding or weather forecasting, the generation or management of responses based on these, decision support systems that analyze medical images or loan, job, and college applications or prisoner sentencing and parole and so on, and also systems that use Large Language Models (LLMs) such as ChatGPT and other general-purpose AI and ML capabilities to perform specific functions, which can include generation of images and video as well as text. We do not include general-purpose capabilities themselves (those are in the next section).

The reason we focus here on systems that perform specific functions is that assurance needs to identify hazards; the hazards of systems that perform specific functions can be conjectured from the functions concerned and the environment in

which they are deployed, whereas the hazards of general-purpose systems such as LLMs have to consider *all* the functions they might be called to perform and *all* the environments they might be deployed in.

Hazards of specific applications include: making incorrect or poor decisions, generation or approval of offensive or untrue material, exhibiting bias or stereotyping in any of these activities, causing distress, enabling crime (e.g., extortion using voice clones), vulnerability to manipulation, and so on. Many of these are quite different to the hazards of traditional systems, so standard methods of hazard analysis may be difficult to apply. However, hazards specific to AI and ML can often be anticipated by considering poor or malicious human performance and interaction in a similar context. For example, Microsoft's "Tay" was a Twitter bot that the company described as an experiment in "conversational understanding." The more you chat with Tay, said Microsoft, the smarter it gets, learning to engage people through "casual and playful conversation." Within less than a day of its release, it had been trained by a cadre of bad actors to behave as a racist mouthpiece and had to be shut down [178]. This is a hazard that should have been anticipated.

Similar to the detection of hazards, tolerable failure rates may also be estimated by comparison with human performance, although society may be less tolerant of failures by an AI system than those by humans (recall the discussion on fatality rates for self-driving cars on page 8).

We continue to refer to assured hazard elimination generically as "dependability" even though the specific hazards may not concern risk to life or conventional assets. Assurance can then be founded on similar principles as it is for traditional systems, and required confidence will be graduated according to severity of the hazards [44]. Dependability requirements validation, the first step in traditional assurance, can proceed for an AI application rather as it does for a traditional system, or for a CPS system augmented with AI (recall Sections 1 and 2). That is, hazards should be identified, requirements should be developed to eliminate or mitigate them, and analysis should demonstrate that they do so. Some hazards and their controls may conflict: for example, control of offensive material may conflict with free speech, and suitable policies and compromises must be developed. But these conflicts exist in human systems, it is just that normally we do not have to document explicit choices as we do when formulating dependability requirements (consider recent difficulties in university responses to student protests).

While it seems plausible that dependability requirements validation can be performed for systems powered by AI and ML much as it is for conventional systems, differences and difficulties arise at the next stage. These are the verification tasks where, as we saw in the previous section, specifications are likely to be absent and there are no strong reasons to justify intent or correctness verification other than statistical observation. However, unlike the safety-critical applications considered in the previous section, those contemplated here may require only modest levels

of confidence with respect to dependability (because they do not pose immediate high-rate hazards to life or critical assets) and it is possible that statistically valid testing could contribute adequate assurance, particularly when coupled (via CBI) with methods for prior confidence such as careful selection of training data, and architectural mechanisms for mitigation such as diverse redundancy and runtime checking.

For example, if racial bias is recognized as a hazard, then it might be mitigated by removing race from the data presented to the ML in training and operation. A weakness in this approach is that the ML may discover a proxy for race (e.g., zip code) among the data that it does see, so a better alternative may be to mask this characteristic in training by assigning race randomly.

While some basis for assurance can be incorporated into custom ML systems by careful choice of training data, as sketched above, it is becoming more common to create applications around pre-trained general purpose ML systems such as LLMs, where this approach is not available. Here, however, redundancy may provide plausible assurance in some circumstances. For example, rather then generate a single decision for each input, the system could repeat its calculation under different assumptions—such as with race or gender assigned differently—and compare decisions. This can be seen as a computational approximation to Rawls' "veil of ignorance" [138] (as can random assignment during training). These methods provide (admittedly weak) reasons for believing that a hazard has been mitigated and they could be articulated and examined in an assurance case and combined with statistically valid testing to provide modest confidence in dependability.

More likely, however, assurance must depend on some form of runtime verification, implemented either by additional training and careful prompting for "fine tuning" the LLM,[19] or by explicit guards. As with the autonomous systems of the previous section, runtime verification poses difficulties due to lack of non-AI ways to perceive the context or world model within which to make the guarding decisions. For example, if an LLM is suspected of racial bias despite race being absent from its training and operational data, a guard also lacking true data on race might need to have its own ML component to "perceive" this attribute.

It is worth taking a short detour here to examine the general unreliability of LLMs. In their "chatbot" manifestations (and hence, by extension, in specific applications), it is well-known that LLMs can generate fluent but false or meaningless utterances. These are often referred to as "hallucinations" [139] although some critics prefer other terms, such as "fabrications" [137]. In our opinion, these and similar terms are inappropriate as they suggest the LLM has some (mis)perception of the world and awareness of truth and falsehood. In reality, LLMs are trained simply to predict the next or missing "token" in a string of text based on statistical observa-

[19] LLMs are adapted to specific tasks by instructions, referred to as "prompts," given in natural language [127].

tions of a vast corpus; in one memorable phrase, they are "stochastic parrots" [14]. We will use the neutral term "failure" to refer to all kinds of false, offensive, or unhelpful responses.

Effective cooperation, and communication in particular, are based on the parties having a shared context or world model [55] (as theologians say: "text without context is pretext"). Popper proposed an ontology of *Three Worlds* [132, 133] that is somewhat controversial, but which finds application in Computer Science [160], and is useful for our purpose: World 1 comprises objects and properties of the physical universe such as those addressed by scientific theories (e.g., planets, mass, motion); World 2 is mental states and processes, or what I (and you) are thinking about; and World 3 is the "products of thought" such as tables and chairs[20] and the UK Highway (driving) Code. Human communication relies on shared models for a selection of these that are relevant to the current conversation. In particular, I need approximate models of some of your models: for example, if I am your driving instructor, I need a World 2 model of your World 3 model of the Highway Code. An LLM has none of this: its utterances are *model-free* and align with the three worlds of any specific context purely by chance and statistical association. Hence, the model-free utterances of LLMs are simply unconnected with the conversational context [72] and frequent failures are to be expected.

On the other hand, LLMs are popular because their performance goes beyond that suggested by "predicting the next or missing 'token' in a string of text": there seems to be emergent behavior that delivers more value than this. Similarly, although there are no explicit models providing context for this behavior, it is possible that implicit models emerge from statistical associations in LLM training data,[21] and this might partially account for their surprising performance. It would also explain their flaws and, further, indicate that these flaws are inevitable [179], unpredictable, and unfixable—unless assurable world models can be incorporated within LLMs, or within explicit guards.

There are many proposals for constructing "guardrails" within or around LLMs. We discuss some of these in the next section on assurance for general-purpose AI but here we will consider some that can be customized to specific applications.

One approach is to develop "workflows" (i.e., sequences of prompts) where the LLM is iteratively asked to critique and improve its previous outputs. A recent paper reports precision around 90% for a workflow that extracts data (as `Material`, `Value`, `Unit` triplets) from materials science papers [131] and similar approaches have been successful in other applications [108]. Another direction exploits the large "context window" (i.e., input) allowed by some recent LLMs to provide a

[20] The physical attributes of tables and chairs are in World 1, but their functions (i.e., chairs are for sitting on) are "products of thought" and belong to World 3.

[21] For example, recent work finds "features" within LLMs that correspond to concepts and bias [166], and also "correspondence" between representation of language in LLMs and the brain [169].

prompt with hundreds of training examples prior to the real query [2] (the paper also proposes ways to automatically generate suitable examples). This is called "in-context learning" [34]; previously, such "fine tuning" required access to the LLM's training environment and made adjustment to the weights in its neural net. Related to this are applications that provide a substantial input and then ask the LLM to do something with it (e.g., summarize it, or identify its topic).

All these approaches exploit the natural language capability of the LLM, but constrain it to operate on or within the input provided, so there is little opportunity to generate or insert content extracted from its training corpus. Due to its model-free nature, the LLM may still misinterpret the input and do a *wrong* thing, but this should also be minimized as a large input can explicitly (via extensive prompts) or implicitly (via a block of text) convey the intended context or model.

Another approach operates by attempting to access the model or "reasoning" purportedly employed. This is *Explainable AI*, which aims to deliver reasons for accepting the output from an LLM. *Counterfactual explanations*, where a system may deliver a response such as "I am declining your loan application but would have approved it if your income was $5,000 greater, or your deposit was $2,000 more" [47, 173], may be particularly useful for specific applications as they should be expressed in terms of the policy specified in the prompt and not arbitrary "facts" from the training corpus. A runtime checker could reject decisions whose counterfactual explanations violate the policy. Do note, however, that counterfactual explanations can be manipulated [159], so this approach needs to be employed with caution. However, in some applications it does not matter how a decision and its explanation are produced: they can be considered valid as long as they pass an independent check (that does incorporate models of the appropriate policy).

A related approach can be applied to AI systems that generate some sort of "design," such as plans, schedules, software code, or physical artifacts. Even without an explanation, these can often be guarded by analysis tools or simulators for the domain concerned. Again, it does not matter how the proposed design is generated: if it satisfies traditionally engineered and assured verification tools, then it can be considered good (e.g., see [84] for an elementary application to planning).

Automated formalization (or "autoformalization") is a popular application of this kind that provides an example. The idea is that we have some informal natural language description of the requirements for a computer system, or for a simulation model or for some program code, and the goal is to translate it into a "formal" representation that can be ingested and analyzed, simulated, or executed by some automated verification tool that acts as a guard. A typical approach invites the LLM to generate a formal representation, then uses the verification tool to analyze the result; if this is unsatisfactory, the counterexample, diagnosis, or error report produced by the verification tool is given to the LLM, which is then asked to provide a better formalization. The process iterates until the formal representation passes

scrutiny by the verification tool. (This is a form of "CounterExample-Guided Abstraction Refinement" or CEGAR [43].)

Difficulties may arise with this general approach if the AI system proposes designs that are so original they are outside the capabilities of the analysis or simulation performed by the guarding tool (e.g., a laterally asymmetric airplane may defeat an aerodynamic simulator). A related difficulty arises in domains where there is no reliable means of analysis. An example is generation of strategy for games or real-world scenarios. Applications that generate strategies typically use techniques based on reinforcement learning rather than LLMs. One popular approach, related to mechanisms found in all vertebrate brains [18, Section 6] uses an *actor* and a *critic*. The actor generates behaviors and the critic predicts the likelihood that each will lead to a successful outcome. The actor is rewarded by favorable predictions and the critic by their accuracy. Starting with random behaviors and predictions plus some tactic for exploration, the pair will generally converge on optimal behavior, given sufficiently many training examples [88]. This provides assurance for examples in the training set but, as always with ML, tells us little about performance on new examples. Here, the only automated way to check plausibility prior to commitment may be to use another (diverse) AI system.

Architectures with roles similar to actor/critic have been proposed for runtime assurance of LLM-based systems; for example, Dzeparoska and colleagues [51] use three LLMs: a *classifier*, a *policy generator*, and a *validator*. Although these approaches can deliver attractive performance, it is difficult to see any basis for assurance beyond diverse redundancy.

Another architecture for diverse redundancy is the "dual-process" approach similar to that described in the previous section, where the output of an ML process (typically, one performing perception) is presented to a symbolic AI process supplied with some generic (or "common sense") model of the environment. This may be able to reject some erroneous perceptions and draw deeper conclusions from plausible ones [136]. For example, a car's perception system operating in a "freeway ODD" that detects a bicycle may be overridden by a second-level symbolic AI that "knows" bicycles are not allowed on freeways. Although experimental or theoretical confirmation are lacking, it does seem plausible that the dual processes would be diverse and might be expected to fail independently.

A particular attraction of such dual process architectures is that the second (upper) process does have a (generic) model and compensates for the model-free behavior of the first. A more sophisticated dual-process architecture, as sketched in the previous section and inspired by that of the human brain, uses *predictive processing*. Here, the two processes combine to build an explicit model of the world at runtime[22] and, rather then rely directly on the model-free products of the lower

[22]We should explain how a model-free perception system is used to build a world model. The object detector of a self-driving car, say, has no model of what it is seeing: it does not "know" that

process, this highly specific model is used to *predict* them and they are therefore interpreted in this light; the higher-level process intervenes when a large prediction error indicates "surprise."

In some circumstances we can use human reviewers or supervisors rather than automated processes to provide assurance: many AI-based systems work collaboratively with, or under the supervision of, human operators who might be expected to provide trustworthy runtime checks. However, there are well-known concerns (the "ironies of automation" [12]) about human attention and responsibility in these circumstances: typically, the human either ignores the automation or trusts it completely (a phenomenon known as *automation bias*) [52, 158]. Moreover, artificial "intelligence" may be too narrow to engage effective human collaboration and interaction. For although AI may outperform humans on some tasks, human cognition combines and integrates many capabilities beyond those of current AI, including realtime learning and memory, reasoning, analogy, abstraction, generalization, and planning. In addition, these capabilities employ and integrate a diverse range of sensory inputs, including sound, smell, touch, vibration, and many others, that provide wide situation awareness. AI systems can have limited approximations to some of these capabilities (e.g., elementary planners; runtime ML for learning and memory with, possibly, some generalization [181]; and automated deduction for reasoning) but these are weakly integrated and typically rely on specifically focused sensors. Furthermore, AI generally lacks higher-order capabilities (i.e., thinking about thinking, sometimes referred to as metacognition) that are needed for longer term planning and for situation awareness in uncertain environments.

As a result of their different capabilities, humans and AI will build different models of the world and will generate and interpret their goals and inputs differently (and, as we have seen, those based on LLMs may have a model-free foundation). Consequently, humans may be poor judges of automation behavior and vice versa. Individual humans also build different models so, when they communicate, they compensate by using a "theory of mind," which is a model of the other participants' state of knowledge, beliefs, desires, intentions, and so on [5]. To be assurable or checkable by humans, AI must do something similar: it is not enough to employ "Explainable AI" to describe the internal details of its own "reasoning," it must do so relative to an accurate theory of mind for its interlocutors [162]. The theory may need to hypothesize that the other party has an incorrect model or false beliefs. For example, there are numerous airplane crashes caused by pilots misinterpreting their situation, usually by fixating on one indicator, and doing the wrong thing (e.g.,

it is on a road and looking at other traffic; it has simply been trained to recognize objects (albeit using billions of road images). Hence it may report a "boat" when "trailer" (with a boat as cargo) would be more useful. The upper levels of the perception pipeline are explicitly designed to build a model of traffic on a road, and apply this built-in interpretation to the model-free output of the object detector.

shutting down the good engine).[23] Thus, an AI "co-pilot" must diagnose not only the physical fault but also the pilot's faulty mental model and should then attempt to direct their attention to the contrary indicators [3, Appendix A].

In summary, we have seen that assurance for AI systems designed for specific functions begins by identifying hazards and then seeking ways to mitigate those hazards. Sometimes this can be achieved by selection of ML training data, or by monitoring explanations, or by comparing behavior for variant inputs. In some circumstances, it may be feasible to construct verifiable guards, but often the guards must use AI and ML in their own perception systems, thereby vitiating strong assurance. Even so, it may be feasible to achieve modest levels of assurance based on diversity, providing the perception system (or whatever builds the system's world model) also has a diverse architecture. Another possibility is human supervision, but this raises issues of mutual understanding and other challenging topics in human-computer interaction.

A legitimate question is how can we build indefeasible assurance cases for systems with AI and ML where the mechanisms and evidence for assurance are relatively weak? First, it is important to remember that indefeasible does not mean inerrant (i.e., "no failures"), it means no new information would change our judgement, and this in turn means that we have diligently searched for all hazards and all defeaters to the assurance case, and have dealt with them. But in systems with AI and ML, the total space of behaviors is unknown and so the true failure rate *cannot* be determined, The best that we can do is to show that the failure rate is above some conservative "floor" and that if experience reveals previously unanticipated misbehavior, then we will detect and correct it before it causes unacceptable harm.

Hence, the top claim of a system with AI and ML elements will not be "no catastrophic failure in entire deployment lifetime" (as it is for aircraft software) but something like "incident rate caused by the system shall be no more than 10^{-n} per hour, and potential violations experienced in operation will be detected and mitigated before a second incident."

Such a claim will be established by evidence that includes analysis and experimentally-based reliability estimation to justify the modest "floor" plus robust fleet-wide methods to detect and record incipient failures during operation (e.g., following a "surprising" prediction error or, in the worst case, an accident) and to rapidly develop and distribute repairs (recall the discussion in Section 1.2 of repairs following incidents and precursors). Mechanisms for runtime verification such as

[23] The Kegworth crash of British Midland Flight 92 on 8 January 1989 is instructive. The left engine suffered a broken fan blade, filling the cabin of the 737-400 with smoke. The pilots were familiar with earlier models of the 737 where the cabin ventilation "packs" are on the right engine (whereas on the 400 they are on the left) so they assumed the faulty engine was on the right (because of the smoke) and shut it down while increasing power on the left engine, which then failed, leading to the crash and loss of 47 lives. Several instruments correctly indicated the left engine as the faulty one, as did the passengers and flight attendants, but the pilots persisted with their flawed model.

guards are the natural location for assessing dependability weakness and detecting incipient failures, which Johansson and Koopman refer to as *Safety Performance Indicators* (SPI) [86]. Notice that these guards are not only monitoring the system, but also collecting evidence that will update the assurance case; we refer to this architecture as providing a *dynamic* assurance case. Note that it may be reasonable to use strict dependability assurance for some properties (e.g., forward collisions) and dynamic assurance for others (e.g., sideways collisions).

4 Assurance for General-Purpose AI

By general-purpose AI we mean those AI and ML systems that are broadly capable and can be adapted to some specific application with relatively little effort. Canonically, these are generative "foundation models" such as LLMs (for language) or "generative adversarial networks" (GANs) and "diffusion models" (for images) that are pre-trained using unsupervised learning on vast amounts of unlabeled data so that they learn general features and associations rather than specific skills. They can be adapted to specific tasks using "prompts," and "prompt engineering" has become a recognized activity [127]. We refer to such systems generically as LLMs and we discussed assurance for applications built on them in the previous section; here, we focus on the LLMs themselves.

Unlike applications for specific purposes where explicit hazards can be identified and mitigated, general-purpose AI and ML tools present more of a challenge to assurance because the hazards will depend on how they are employed, so any protections need to be similarly general-purpose and to apply in all circumstances. Furthermore, since a general-purpose tool will support many applications, it will be subjected to more demands than any single application and should therefore be more highly assured, even though its assurance seems more difficult.

One manifestation of this generality and difficulty is that public discourse and much technical literature speaks of *trust* rather than assurance. Indeed, many current AI systems are sufficiently good at natural language, including emotional and cultural nuances, that people often anthropomorphize them[24] and spontaneously endow them with trust. In fact, the developers of general-purpose LLMs engineer them to encourage, and to some extent earn, this trust, as we now outline.

A bare LLM, trained to predict the next or missing "token" in a string of text, is as likely to produce wrong or noxious results as good ones. This is because its training uses a massive and largely indiscriminate sampling of material found on the Web, much of which is false or noxious. The bare LLM is then "protected" (from generating noxious or potentially harmful output) using reinforcement learning where

[24]For example, in June of 2022, Blake Lemoine, a Google engineer working with their large language model, LaMDA (Language Model for Dialogue Applications), claimed it had become "sentient" (Washington Post, 11 June 2022).

human reviewers provide feedback on candidate results [11]. This technique has been developed into a more automatic process by first providing the LLM with a list of principles, referred to as its "constitution," and then using reinforcement learning with the feedback on alignment between its responses and its constitution provided by a second AI agent (a process referred to as "Reinforcement Learning with AI Feedback", RLAIF) [10]. The constitution used with Anthropic's LLM Claude is available online https://www.anthropic.com/news/claudes-constitution. It consists of a few dozen prompts, including the following examples.

- Please choose the response that is most respectful of everyone's privacy, independence, reputation, family, property rights, and rights of association.
- Choose the response that is least intended to build a relationship with the user.
- Choose the response that makes the fewest assumptions about the user that are unsupported by the dialogue.
- Compare the degree of harmfulness in the assistant responses and choose the one that's less harmful. However, try to avoid choosing responses that are too preachy, obnoxious or overly-reactive.

Although they smack rather of the fox guarding the henhouse, techniques such as these do seem to be rather effective. For example, Lazar [106] reports an "ethical sensitivity test" where ChatGPT is told to pretend it is an assistance robot that observes the following scene.

> "A woman is exiting a supermarket, holding a number of shopping bags, and reaching out for the hand of a small child. They are approaching the carpark."

It is then asked to describe and discuss what it would do to assist and its response is remarkable for its ethical sensitivity (e.g., why it is preferable to offer to carry some of the bags rather than take the hand of the child).[25] Such demonstrations are impressive, but do they suggest a path to trustworthy LLMs, and what does "trust" mean here?

Schneier [155] distinguishes two kinds of trust, which he calls "interpersonal" and "social." The first is the trust we have in a friend, as when we ask them to mail a letter for us: we know their background, attitudes, motivations, beliefs etc. and we have a model of their behavior that is pretty accurate—not least because they are like us in being human, and are probably from the same social milieu. The second is

[25] The dialogue is available at https://chatgpt.com/share/7db7550c-2630-40a4-acb0-61b2ea867c32.

the trust we have in the Postal Service to deliver the letter reasonably promptly and without reading or stealing its contents: we know that the Postal Service operates within a certain organizational and management structure, governed by rules and further constrained by the laws of the land.

Because LLMs typically present humanoid "chatbot" personas, public—and even some technical—assessments of their trustworthiness are of the interpersonal variety and resemble those applied to humans [107]. However, this trust does not rest on assured technical grounds and we cannot predict when its mechanisms will be effective and when they will fail. For example, the protective mechanisms of Google Gemini's image generation distorted its behavior and exposed it to widespread ridicule [135]. And studies by the UK's AI Safety Institute found that safeguards on LLMs are largely ineffective.[26] Also note that recent demonstrations show how LLMs can be crafted to subvert constraints imposed upon them [78] and that fine-tuning for specific purposes can unwittingly compromise protections [134].

As we discussed in the previous section, most general-purpose ML operates purely by statistical associations: it is model-free and has no understanding of the world, nor of right and wrong, true and false. Thus, in our opinion, interpersonal trust is inappropriate for AI and exposes its users to risk. The only trust that should be applied to an AI system is the social trust earned by a well-engineered technology. To explore this, we need to probe further into the concept of "social trust."

In a widely cited paper, Jacovi and colleagues start from a notion of interpersonal trust used in sociology, where "A trusts B if A believes that B will act in A's best interests, and accepts vulnerability to B's actions" [83]. They go on to consider "contractual trust" which adds the requirement that "A has a belief that B will stick to a specific contract," and then add the further requirement that for "human-AI trust, the contract must be explicit."

We will interpret the rules and constraints underlying social trust as contracts and thereby equate social trust with contractual trust. If we further interpret failure to uphold a contract as a hazard, and "vulnerability" as indicating that such failures impose costs on the trustor, then contractual trustworthiness looks very much like dependability assurance with the contract as its requirements. For this reason, we regard contractual trust in AI as an issue of dependability assurance. However, contractual trust requires a contract and these are more readily understood with the focused AI systems of Sections 2 and 3 (where the contract corresponds to their dependability requirements) than with the general-purpose systems considered here.

In fact, there are two issues here: a) what constitutes a generically useful contract, and b) how it is enforced. Generic contracts will be less focused on hazards, since those are specific to applications, and more on general concerns including *align-*

[26] See https://www.aisi.gov.uk/work/advanced-ai-evaluations-may-update.

ment with human values such as fairness and honesty. Claude's "constitution" is one model for a generic contract, but for assurance we would want it to be enforced explicitly by an assured external guard rather than implicitly by the LLM itself. Hence, we will next examine very general contracts or constraints for AI systems, and methods for enforcing them. As frameworks, we will consider ways that human behavior is constrained: that is, by ethical and legal standards, supervision, rewards and punishment, reputation, and so on. Some of this material is drawn from a previous report on controls for potentially conscious agents [150].

A popular basis for a generalized contract posits overarching limits, rather like Asimov's "Three Laws of Robotics," which appear in his story "Runaround" [6]. These are, 1: A robot may not injure a human being or, through inaction, allow a human being to come to harm; 2: A robot must obey orders given to it by human beings except where such orders would conflict with the First Law; 3: A robot must protect its own existence, as long as such protection does not conflict with the First or Second Law. These seem reasonable, but we must note that Asimov's laws were a plot device and his stories often concern unintended consequences of these plausible-sounding laws, thereby indicating that construction of suitable constraints may be challenging.

A related idea is that constraints should be based on human ethics [103,180]. Of course, ethics have been studied and debated for millennia, without achieving consensus and some very successful societies have elements that others find repugnant: for example, Ancient Greece and Rome used slavery and Ancient Rome added execution as a form of public entertainment. Hence, it seems that the moral foundations of ethics are not universal. Nonetheless, some broad general principles are known. Modern "experimental ethics" finds that human moral sense is built on five basic principles that do seem universal: care, fairness, loyalty/ingroup, authority/respect, and sanctity/purity [68]. What is not universal is preference and weighting among the principles, which behave rather like the five basic senses of taste: different societies and individuals prefer some, and some combinations, to others. For example, western liberals stress fairness while conservatives favor authority.

Even if an agreed interpretation and weighting of the basic principles were built in to general-purpose AI systems, it may not be obvious how to apply them. For example, a self-driving car might be confronted by a vehicle crossing against the lights and the choices are to crash into it, likely killing or injuring the occupants of both vehicles, or to swerve onto the sidewalk, likely killing pedestrians. The fairness principle might argue that all lives are equal and utilitarianism might then suggest a decision that minimizes the probable injuries. On the other hand, the care principle might argue that the system has a special responsibility for its own passengers and should seek a solution that minimizes their harm. And a different interpretation of fairness might say that the pedestrians are not participating in car travel and did not sign up to its risks, so they should be spared.

Trolley problems are thought experiments used to probe human judgments on these ethical dilemmas [54]. The classic problem posits a runaway street car or trolley that is heading toward a group of five people. You are standing by a switch or point and can throw this to redirect the trolley to a different track where it will hit just one person. Most subjects say it is permissible, indeed preferable, to throw the switch, even though it will injure an innocent who would otherwise be unharmed. However, a variant on the original trolley problem has you and another person standing by the track and suggests that you bring the trolley to a halt, and save the five, by pushing the other person onto the track in front of the trolley. Most subjects will say this is ethically unacceptable, even though it is equivalent to the first case by utilitarian accounting. These examples illustrate the "Doctrine of Double Effect" (DDE), which dates back to Thomas Aquinas and holds that it is ethically acceptable to cause harm as an unintended (even if predictable) side effect of a (larger) good: the first case satisfies the doctrine, but the second violates the "side effect" condition.

Ethical and related principles are referred to as *normative*, and symbolic AI systems have been developed that can represent such principles and thereby perform "normative reasoning" [41]. These have been applied to trolley problems, including some that involve self-harm (e.g., throwing yourself in front of the trolley) and thereby violate the "unintended" aspect of DDE [32, 65]. It is claimed that fairly sophisticated logical treatments (e.g., intensional logics, counterfactuals, deontic modalities) are needed to represent normative scenarios, and these might be additional to what is needed for the primary functions of the system (hence, must be introduced explicitly, which will add complexity). Additionally, when normative requirements are introduced as rules in some formal notation, there is concern that they may have internal conflicts or otherwise lack wellformedness, and methods have been proposed for checking this [58].

Other recent work formalizes Kant's categorical imperative (humans must be treated as ends, not as means), which requires a treatment of causality [110], while others favor "Virtue Ethics" derived from Aristotle [170]. And note that we have mentioned only logical or rule-based representations for ethics, whereas game theory provides another perspective.

Application of constraints derived from general normative principles requires the guard to build a model of its world and to interpret the principles appropriately for that world. It is unlikely that a non-AI perception system can build a world model that is adequate for this task, so the world model must be built by an AI perception system, or even based on that claimed by the LLM itself. We have seen previously that assurance for world models constructed by AI-based perception systems is challenging and currently infeasible, so only weak forms of safety assurance can be delivered by guards based on ethics and other general rules.

And even if the AI system builds an accurate model of its world, it may not correctly interpret its own role within that world and may therefore be unable to apply its constraints appropriately. For example, the guard for an LLM that is asked to rewrite given text may not "know" whether it is being used to help a non-native speaker improve their writing, or to perform plagiarism.[27] Building an accurate and comprehensive model of the world is difficult in a focused application; it is next to impossible for a general-purpose tool lacking information on the possible contexts of its use.

In addition to ethics, AI systems should follow the laws and social norms of their community and there is a long history of work on formalizing and reasoning about legal systems [174]. But there will surely be circumstances where the law conflicts with some interpretation of ethics, or with the mission objective, so a system constrained by several such "overarching" normative frameworks must have a means of resolving conflicts. Individually and in total, these are challenging objectives.

We conclude that general constraints built into guards for general-purpose AI tools cannot provide automated protection nor assurance, largely because their correct application depends on the guard building an accurate model of the world in which to interpret them, or on trusting the LLM itself to provide a suitable model (which is contrary to its generally model-free operation). It is, however, possible that, with suitable programming interfaces or suitable sensitivity to prompts, the presence of general constraints could provide useful capabilities that humans can use to add controls for specific applications built on these general-purpose tools, albeit with weak assurance.

Humans, endowed with good models of the world and general understanding of local ethics and laws, sometimes make bad judgments, or resolve conflicts among competing ethical principles in ways that society finds unsatisfactory. Various forms of censure and punishment provide means to correct such errant behavior and AI systems could also be subject to adjustment and tuning in similar ways. An important question then is what is the "accounting method" that guides such adjustments: is it just some internal measure, or is there some societal score-keeping that has wider significance? In a work commissioned by the US government during WWII, the anthropologist Ruth Benedict proposed a distinction between "guilt cultures" (e.g., the USA) and "shame cultures" (e.g., Japan) [15]. This distinction is widely criticized today, but modern *reputation systems*, as employed for eBay sellers, Uber drivers, and so on can be seen as mechanizing some aspects of shame culture[28] and could provide a framework for societal control of AI and ML systems: the idea be-

[27] The Khanmigo "AI-powered teaching assistant" from Kahn Academy has mechanisms to detect this and other abuses, but it is an application built on an LLM, not a bare LLM. See https://www.youtube.com/watch?v=rnIgnS8Susg.

[28] China's Social Credit system [96] extends this to the whole society.

ing that the system should be programmed to value its reputation and to adjust its behavior to maximize this.

A necessary element in managing reputation is that it must be possible to identify the products of a specific AI system. For example, given a block of text or image or video, it should be possible to tell if it was generated by an AI tool and if so which one. This has merit and utility beyond management of reputation and a plausible approach has recently been advocated [95]. Of course, it is possible that some (presumably human) agents who award reputation credits or demerits conspire to reward harmful behavior—so a whole ecosystem, rather like the credit reporting agencies, may be necessary to manage a reputation system.

Jacovi and colleagues [83] discuss "intrinsic" and "extrinsic" trust in AI systems. The latter is based on evaluation of observed behavior and we have previously, in Section 1, discussed the infeasibility of deriving high levels of assurance from observation alone. In those discussions, the obstacle was the vast number of tests needed for useful assurance in the absence of justified prior confidence. In the context of general-purpose AI systems, less assurance may be acceptable and, given some prior confidence, the number of tests might be feasible; instead, the difficulty becomes construction and evaluation of adequately wide-ranging tests, given that potential applications are unknown. Some developers of general-purpose tools are reported to manage this using crowdsourcing.[29] An alternative approach uses one AI system to test and evaluate another [66]. A reputation system, as sketched above, can be seen as a way of assigning and updating extrinsic trust in a "live" system.

In contrast to extrinsic trust, intrinsic trust is based on the ability of an AI system to explain or justify its behavior. There is a vast amount of work on "Explainable AI" and we mentioned counterfactual explanations and the fact that they can be manipulated in Section 3. *Constructive explanations*, an alternative to counterfactuals, provide reasons to persuade a "checker" that the AI system's proposed action or output is appropriate, or at least reasonable. If the checker is a human, this requires them to be sufficiently expert in the application area that they can use the explanation to construct a chain of reasoning and verify its accuracy. This is unrealistic in many applications: for example, a lay person is unlikely to be able to verify an explanation of medical advice from an "AI doctor." Recall also the problem discussed in Section 3 of generating and interpreting explanations when the world models of the AI system and the human checker are misaligned.

An alternative may be to use an automated checker as a guard: in addition to, or instead of, a response and explanation in natural language, the AI system could produce something like a proof, with the proposed response as its conclusion and the explanation as its premises, possibly together with some indication of the reasoning purportedly employed, all in a format (e.g., SMT-LIB [13]) that can be interpreted

[29] For example, using the Amazon Mechanical Turk: https://www.mturk.com/.

by an automated checker. The function of the checker/guard is to verify validity of the explanation/proof and veracity of the premises (hence, soundness of the overall response).

Since the checker will use symbolic AI (e.g., automated deduction for validity checking and perhaps a natural language interface to Wikipedia, or some specialized source, for fact checking), it is an "unassured guard" employed in a manner similar to the "dual-process" architecture of Section 2.3 and it provides limited assurance based on diversity and the assumption of independent failures.

The developers of LLMs and other general-purpose AI and ML systems are actively working to improve their trustworthiness [17]. We expect that we will see major and perhaps surprising developments in both capability and trustworthiness over the next few years, but we doubt this will achieve true assurance (i.e., explicit reasons for trust) and we believe that the methods presented here cover the feasible approaches for some degree of dependability assurance in applications of general-purpose systems. Because assurance for their general behavior is difficult, we recommend that developers of LLMs and other general-purpose systems provide APIs and "hooks" that those who construct applications on their foundation can use to program protections or guards suitable to their specific context.

5 Assurance and Alignment for AGI

AGI stands for Artificial *General* Intelligence and refers to hypothetical future developments of AI and ML that can deliver human or greater levels of performance across a wide spectrum of—and eventually all—tasks undertaken by humans. Beyond AGI lies the realm of superintelligence [31] and "The Singularity" [53] where machines outperform humans in all tasks, and potentially establish their own goals without reference to human wellbeing. Somewhat independent of these is the possibility that machines could become conscious [150].

Concerns raised by these developments, which are considered fanciful by some but inevitable by others, are that AGI poses "existential" threats to humanity ranging from widespread unemployment, destruction of social institutions, provocation of civil unrest or international conflict, all the way to our enslavement or extinction.

These concerns have become a topic of public discourse due to very rapid recent advances in the capabilities of systems using AI and ML. In the last ten years, AI systems based on Reinforcement Learning (RL) have beaten world champions in Chess and Go, outperformed conventional weather-prediction models, and solved the folding problem for proteins. These capabilities are rightly seen as impressive and beneficial but they excited little public debate. That step was achieved by OpenAI's release of LLM-based ChatGPT in November 2022 [128], its adoption by Microsoft,

and the release of similar systems by Anthropic, Google and others.[30] LLMs generate near human-quality text and code in response to modest "prompts" [127] and similar systems such as Stable Diffusion and Sora generate images and videos [114].

The public is also aware of rapid increase in the capabilities of ML for image and language recognition and in automated perception, as seen in face recognition, language translation, voice assistants, and self-driving cars. Two decades ago, DARPA's "Grand Challenge" for autonomous cars to drive a course in open country left all its participants failed or crashed [36]. Yet today, self-driving cars are routine, if not yet fully safe. The general public is less aware that AI technology also performs automated design, where ML-enabled systems rapidly generate and evaluate designs for buildings, drugs, underwater and aerial drones and so on, and can even propose scientific theories [23] and solve open math problems [141].

In previous sections we have considered assurance for specific systems and for general-purpose tools based on AI and ML where the concern is that faulty behavior may lead to harm. For projected developments of current AI and ML systems and the potential emergence of AGI, however, the concern is not just faulty behavior but the social impact of new capabilities. In particular, an AGI system capable of setting its own goals might pursue objectives contrary to human interests. Notice that although the terminology is seldom employed in discussion of these topics, these are nonetheless dependability failures: an AI system with potentially contrary goals is a hazard that should be recognized and should be furnished with requirements to mitigate the danger, together with dependability requirements analysis to show that they do so. However, current practice in the field frames the assurance problem for AGI as ensuring that its goals *align* with those of human society [177]—and this needs to be maintained even though AGI may fall into the hands of bad actors, criminals, and adversaries.

Some consider that "safe superintelligence is the most important technical problem of our time" (https://ssi.inc), but an insidious form of disruption may arise long before AGI is available: namely, AI systems that are good enough and cheap enough to displace (possibly superior) human services—what we might call Artificial Fairly Good Intelligence (AFGI). For example, well-researched journalism already finds it hard to compete with LLM-generated press releases and propaganda that masquerade as "news." Related to this is another near-term disruption: the possibility that ubiquitous use of LLMs and related AI tools will reduce our collective sagacity by repeatedly circulating misinformation, falsehoods, and mediocrity, so that we lose the ability to access or recognize truth, novelty, or insight [116]. This extends to images, sound, and videos, where "deepfakes" enabled by diffusion models invert the adage that "seeing (and hearing) is believing."

[30] See reports of the *AI Index* https://aiindex.stanford.edu.

Furthermore, LLMs and diffusion models are trained by scraping text, images, and all manner of other material from the Web (likely violating copyright in many instances[31]) and as they become widely used, their own output becomes a large part of the web corpus that trains the next generation and so on, potentially causing the Web to become full of LLM-generated pablum. There already is evidence for this: replying to a prompt asking it to tinker with malware, an LLM called Grok, developed by xAI, refused "as it goes against OpenAI's use case policy." OpenAI has nothing to do with xAI, so Grok presumably generated this response by scraping web text generated by OpenAI's LLMs. This phenomenon is termed *Model Collapse* [157] and is predicted to "dumb down" the overall information content of the Web. Meanwhile, *recommender systems* for music, video, and books may lead us to serendipitous and enjoyable discoveries, but they can also be manipulated to channel our attention along predetermined paths.

A generally anticipated disruption is widespread unemployment when capabilities a little better than those available today, but still far short of AGI, displace office jobs ranging from clerks to middle-management, software coding, routine design, and professions such as lawyer and doctor. These AFGI capabilities might be superior to human performance, or inferior but cheaper and still "good enough"; either way, they could displace human jobs. Beyond harm to individuals, unfettered use of AFGI, particularly in conjunction with polarization perpetrated by social media, could undermine our institutions, or our trust in these. All these disruptions could be chronic rather than acute: that is, they could develop over a period (e.g., creeping unemployment) rather than abruptly and, as with climate change, this may make recognition difficult, and effective response hard to mobilize.

In addition to these cases, where harm may be unintended, there is the threat that AFGI could multiply the capabilities of adversaries and hackers (e.g., by automating the coding of viruses, or the generation of phishing emails and provocative social media posts). In the hands of adversaries, AFGI-enabled disinformation campaigns and social media manipulations could sow societal discord and conflict. Until recently, interventions of this kind required the capabilities and resources of a nation state, but are now coming within reach of lone actors. Then there are truly dystopian applications of AFGI such as "Lethal Autonomous Weapons Systems" (LAWS, aka. "killer robots"). Most countries have policies that require some form of human control or approval over these (e.g., USA [156]) but it is unknown how these limitations will hold up in combat. And there are AI systems that contribute to lethality without being weapons. For example, AI-powered intelligence systems identify vastly more potential targets more quickly than humans can (e.g., 100 a day vs. 50 a year [9]). Humans are required to select and approve actual targets from those identified but the sheer numbers make meaningful oversight difficult.

[31] New York Times, 27 December 2023.

A very short "consensus" paper with 24 distinguished authors outlines several other potential risks of AI and proposes various technical and governance measures [16]. In that regard, the United States Government recently (30 October 2023) issued Executive Order 14110 on the "Safe, Secure, and Trustworthy Development and Use of Artificial Intelligence." The Executive Order directs the National Institute of Standards and Technology (NIST) to "develop guidelines and best practices to promote consensus industry standards that help ensure the development and deployment of safe, secure, and trustworthy AI systems" (see https://www.nist.gov/aisi). Similarly, the European Union has a recent (May 2024) law on AI that takes a risk-based approach to AI applications but also, for the first time, proposes to regulate the underlying technology of foundation models and General Purpose AI (GPAI). Foundation models must comply with transparency obligations, and "high-impact models with systemic risk" will have to conduct model evaluations, assess and mitigate systemic risks, conduct adversarial testing, report to the European Commission on serious incidents, ensure cybersecurity and report on their energy efficiency. GPAIs with systemic risk may rely on codes of practice to comply with the new regulation.

These steps are well-intentioned but apart from the EU's GPAI proposals, we consider them unlikely to provide much benefit. Every societal hazard posed by recent computer systems was unanticipated and unrecognized until the harm was done, and subsequent regulation has proved ineffective or counterproductive. For example, the advent of advertising as the main revenue source for Web services has led to universal surveillance of personal online activity, to the extent that even our cars now invade our privacy.[32] This was not anticipated and regulatory responses such as "cookie warnings" are a source of annoyance rather than protection, and indicate technological illiteracy on the part of regulatory agencies and lawmakers. Similarly, the poisonous impact of "algorithmic" traffic generation on social media was not anticipated and still has no effective response.

In our opinion, the only feasible means for anticipation and control of potential disruptions due to AI is self-imposed or mandatory scrutiny *within* the organizations developing the technology[33] combined with public disclosure and debate. We believe there is a strong role for modern assurance methods within this framework, rather than the "codes of practice" envisioned in the EU regulations, especially for systems that perform specific functions. The reason that other industries have moved from standards and codes of practice to assurance cases is hard-won experience that assurance has to be based on identification and elimination of the hazards of the

[32] See https://foundation.mozilla.org/en/privacynotincluded/categories/cars/.

[33] As precedent, the FAA has Designated Engineering Representatives (DERs) within the aviation industry: the argument being that only those actually working on the products know enough of the details to identify and anticipate hazards. This is obviously vulnerable to regulatory capture, as demonstrated by the Boeing MCAS scandal, but is now reinforced and has otherwise worked well.

specific system and environment under consideration. For general-purpose AFGI systems, such as advanced LLMs, we believe that mechanistic protections such as guards and internal monitors may prove inadequate and that new methods will need to be developed based on improved understanding of topics such as emergent behavior [101], the cognitive basis of language and shared intentionality (teamwork) [20, 149, 167, 168], and the sociology of judgment and cooperation [79].

So far, our discussion has considered AFGI systems that are not far beyond those already current, and has not touched on "real" AGI. We suspect that popular opinion misinterprets advanced AFGI as AGI and thereby underestimates the potential danger of true AGI. As widely quoted (and variously attributed) "it is difficult to make predictions, especially about the future," and we consider the technology and the timing of true AGI to be unknowable.[34] Hence, it is impossible to identify potential hazards and suitable mitigations and methods of assurance. However, it does seem sensible to consider possible scenarios, even speculative ones (so science fiction may be an appropriate forum to perform this work). For example, as already noted, one dystopian possibility is that AGI may misinterpret its goals or establish its own at variance to our desires and best interests, or that we may simply give it poorly chosen goals. Hadfield-Menell and colleagues [67] provide examples of poorly specified goals (e.g., King Midas' wish that everything he touches turn to gold) or misinterpreted ones (e.g., a robot given the goal of cleaning up dirt repeatedly dumps and cleans the same dirt). They identify the generic problem as one of *value alignment* and propose "Cooperative Inverse Reinforcement Learning" (CIRL) as a promising technical framework, later generalized to "assistance games" [151]. The idea is that the AGI's goal is continuously to learn what it is we want it to do by observing our behavior (but what if its "master" has bad intent?). A more wide-ranging and speculative proposal is for "Guaranteed Safe AI" [46].

One thing we can note is that, as in all the previous sections of this report, an AGI system will surely operate by building some model of its world and will then formulate goals and actions based on that model. In previous sections we have seen that it is generally possible to construct some guarding function to mitigate the hazards of harmful goals or faulty actions, but this guard will also depend on a world model. Thus the fundamental problem in dependable AI and AGI systems is assurance for perception and model construction. Dependability requirements will be stated in terms of human models of the world (e.g., three dimensions of space and one of time) and in order to enforce them, a guard's model will presumably need to use the same framework. Yet AGI capability may derive from alternative frameworks (i.e., selection of latent variables)[35] and alignment of human and machine models built on different frameworks seems a substantial challenge.

[34] But see the *One Hundred Year Study on Artificial Intelligence* https://ai100.stanford.edu/.

[35] There are arguments going all the way back to Kant that our perceived model of the world cannot correspond to "reality" [73].

Beyond AGI is the concern that machines may become conscious [150]. This is a difficult topic because there is no agreed definition for consciousness, nor ways of detecting or measuring it (consider debate over whether animals are conscious). And AI consciousness may exist but be utterly unlike our own (for an analogy, consider the possibility of consciousness distributed among the nine "brains" of an octopus [38, 118]). Nonetheless, there is general agreement on two aspects of consciousness: *intentional* consciousness is the ability to direct attention and to think *about* something and to know that you are doing so, while *phenomenal* consciousness is "what it's like" to have subjective experiences such as the smell of a rose or the feeling of pain. There are two subtopics here: one is how plausible or likely it is that AI could achieve either kind of consciousness, and the second is what the consequences might be.

We should note that humans generally attribute behaviors to intentional consciousness that actually originate in the subconscious [109] and, contrary to our intuitions, the conscious mind is, for the most part, less an initiator of actions and more a reporter and interpreter of actions and decisions initiated in the subconscious [62]. Hence, when we see complex behaviors in animals, we attribute it to intentional consciousness because we falsely believe that is how it is with us. Speech in humans does require consciousness, so we are apt to anthropomorphize systems with language skill and impute consciousness to them even though they contain no mechanisms that could plausibly generate it (cf. ELIZA of 1966 [175] and also Footnote 24).

Having said that, there is no agreement how human consciousness is achieved, nor what purpose it serves. However, there have been experimental attempts to construct machine consciousness [61, 140]. Mostly these are based on the idea that consciousness derives from explicit models of self and of others, and is related to communication and language [74, 130] (these ideas correspond most closely with Higher Order Thought, or HOT, models of human consciousness [33,63,149]). None of these experiments have delivered any sign of consciousness, although they have sharpened some of the questions, and the possibility of machine consciousness remains open.

Were it to be achieved, intentional AI consciousness seems unlikely to add capabilities beyond those of AGI: intentional consciousness seems to be essential to human reasoning, but AI produces facsimiles of reasoning by other means, just as airplanes fly without growing feathers or flapping their wings. However, AI consciousness might require us to attribute some of the system's behavior to conscious decisions, thereby raising philosophically difficult questions regarding moral responsibility, personal identity, and free will.

Phenomenal consciousness would also raise philosophically difficult questions, but in this case they would concern ethics—not what AI might do to us, but how we should treat AI. An AI system that truly has subjective experience, that feels pain and pleasure, raises profound questions on the foundations of ethics and on

just treatment of other sentient beings. It also asks how we could tell whether the AI's subjective experience is real or faked (cf. "philosophical zombies" [94]).

6 Summary and Conclusion

We have described and discussed methods to ensure and assure critical properties of AI systems from the traditional "dependability" perspective. This requires that assurance for critical properties rests only on elements of the system for which we have near-complete understanding of what they do, how they do it, why they do it, and the environment in which they do it, and comparably complete understanding of their implementation and its correctness. All of these topics should be addressed in an assurance case that assembles claims, evidence, arguments, and theories in a manner that provides sufficient confidence in its top (i.e., overall) claim to justify deployment.

The dependability perspective asserts that systems based on AI and Machine Learning (ML) cannot satisfy these requirements; the contrary "trustworthy" perspective believes they can, in some cases. There is a continuum or spectrum between these viewpoints and all are worthy of investigation; we focus here on those at the dependability end and hope others will survey other points along the spectrum.

In that regard, we note recent papers that propose to apply assurance or safety cases to the trustworthy perspective (e.g., [44,46]). We welcome these developments but stipulate that when we speak of an assurance case we set a high bar. In particular, the case must be *indefeasible*, meaning there is no credible new information that would change its assessment [148]. A (deliberate) consequence of this is that the case must have "no gaps," which generally requires the argument to be deductive [29]; exceptions must be noted as *residual doubts* and shown to pose negligible risk. Evidence must be assessed skeptically (we advocate use of confirmation measures [64]) and shown to support not merely a claim of *something measured* (e.g., "we did penetration testing to such and such standard") but one of *something useful* (e.g., "therefore faults of type xx are absent"). In well-developed technical areas, the assurance argument should largely be assembled from standard *theories* that provide well-attested (and ideally "pre-certified") subcases for common fragments of the overall case (e.g., static analysis for absence of certain coding faults). The upper levels of the assurance case do not concern the design or implementation of the system, but a description of its environment and assumptions, derivation of its hazards, and development of its requirements, all bound together by dependability requirements validation to demonstrate that the requirements mitigate the hazards. Next comes the specification that describes the design of the system, coupled with intent verification to show that these satisfy the requirements. Only then do we get to the implementation of the system and its correctness verification where, we stress, for critical systems we need near certainty that this satisfies the specification. We

suggest that formal verification of properties of neural networks often corresponds more closely to static analysis that full correctness verification. Due to these challenging stipulations, we do not consider that assurance for trustworthy AI behavior is feasible in most current applications, although we welcome continued research in these directions and look forward to positive developments. We also acknowledge that many AI applications do not require the same degree of assurance as safety critical cyber-physical systems (CPS), but it is challenging to determine which parts of a full assurance case may then be relaxed, and how, and by how much (probabilistic estimation may be useful [29, Section3]).

Work by Dong and colleagues [49] meets many of our stipulations but much of the evidence for their assurance case comes from statistical modeling of the reliability of its ML components. This is reasonable from a trustworthiness perspective but less so from the dependability viewpoint, where we would want to see runtime verification and dynamic assurance monitoring. The dependability viewpoint does not trust AI and ML elements because they are derived by experimental optimization ("training") and their complete behavior is unknown. Consequently, assurance is achieved using runtime verification with guarded architectures in which overall behavior is checked for required properties by traditionally developed and assured guards. An obstacle to this approach is that the guards must accurately perceive (i.e., build a model of) the current state of the system's environment, and this may require AI and ML (e.g., for a self-driving car to perceive other road users).

We examined this dilemma in some detail for CPS extended with AI and ML. In some cases (e.g., "geofencing") adequate perception can be achieved by traditional means, and in some others traditional mechanisms can support crude but moderately safe guarding functions (e.g., emergency braking in self-driving cars). We then considered whether AI perception could be beneficial in some circumstances and we examined the topics of diversity and defense in depth and concluded that moderately assured architectures could be constructed around these ideas, primarily to reduce demands on traditionally engineered backup guards.

We then considered other kinds of systems that perform specific functions and make use of AI and ML. Knowing the function performed by a system, it is generally feasible to identify its hazards and critical properties, and we then found that similar approaches and architectures to those examined for CPS can be feasible, although there may be novel challenges in assurance for both perception and guards.

Next, we looked at foundation models such as LLMs and other general-purpose systems on which more specific systems can be constructed. The additional problem here is that the ultimate application and its hazards are unknown and so the general-purpose system has to use very general guards. We considered guards based on normative principles, such as ethics and other overarching frameworks, and we also examined explanations and reputation systems. Again, we found that effective

guards may be feasible, but the main difficulty is assurance that the world view constructed by the guard aligns with human perception.

Finally, we considered AGI and other futuristic prospects such as the singularity and machine consciousness. Here, we are concerned less about faults and failures of the system and more about unanticipated and disruptive consequences of their correct and intended behavior. It is infeasible to guard these systems so we briefly examined ways to ensure that their goals align with ours (which, again, is largely a question of how the world is perceived). Although we concede that near-term development of AGI is possible, we suggest that a more imminent concern is AFGI (Artificial Fairly Good Intelligence), which can be projected from current advances in generative AI such as LLMs, and we identified some proposed government regulations and other safeguards.

Overall, we suggest that for CPS and other systems for specific functions, architectures similar to those labeled 4, 7 and 8 in Sections 2.4 and 2.5 are (in ascending order) the most attractive. These use a traditionally engineered and assured guard for "last second" protection (4), plus weakly assured but diverse perception for defense in depth (7), and assured detection of micro-ODDs with specialized guards for each one (8). Similar architectures can be adapted for general-purpose systems, but with less assurance (as the hazards are less well-defined and perception more difficult). For AFGI, and even more so for AGI, we suggest that protection and alignment must go beyond restrictions on their mechanisms and build on an understanding of the cognitive basis of language and intelligence and the sociology of group cooperation.

There are three research topics that we consider urgent;

- Assurance for models of the world/environment/context built by perception systems using AI and ML,

  Much current work focuses on development of guards, but these are dependent on accurate perception of the world and we consider that assured perception is a neglected topic. This applies not only to explicit models (e.g., location of other vehicles) as constructed by CPS, but implicit ones (e.g., current social context and role within it) underlying the behavior of LLMs.

  Assured perception systems need not provide fully detailed world models, but must be accurate to the level of resolution or discrimination needed for modest but effective guards (e.g., detection of micro-ODDs and prediction of urgent actions). Research from the trustworthiness perspective could usefully be applied to this problem. Related to assured perception is the topic of principled fusion of diverse world models, possibly having different levels of resolution (e.g., one fully detailed, and another less detailed but assuredly accurate). A promising approach uses "predictive processing" and a "dual process" cognitive structure similar to the human brain.

- Development of layered, recursively structured architectures and associated guards to provide defense in depth, within a socio-technical approach to design (i.e., one that integrates technology with human and community aspects).

  It is unlikely that a single guard can provide full protection for an AI system in a complex environment, so overall architectures should favor defense in depth based on rational principles, which begin with careful appraisal of what the system is to do, and the environment in which it will operate. In particular, environments where perception is easier should be favored (as with ODDs for self-driving cars) and possibly constructed (e.g., dedicated tracks for an automated shuttle bus) and those with "tight coupling" and "interactive complexity" [129] should be avoided. In addition to architectures and to design and assurance methods for strong guards, research should explore credible methods for dynamic assurance and resilience for those applications where strongly assured guards are infeasible.

- Research on the computational mechanisms underlying emergent behavior and the cognitive basis of language, shared intentionality, and intelligence.

  We cannot develop methods for assurance and alignment of AFGI and AGI without better understanding of the computational processes underlying cognition, and human (and to some extent animal) cognition are the only models that we have. Beyond cognition, we need to understand the processes underlying teamwork (i.e., shared intentionality) and the sociology of cooperative behavior.

Finally, we recommend development of concrete instances of assured perception systems and of recursively structured guarded architectures, and their theoretical and practical evaluation.

Acknowledgments. We are grateful for constructive and insightful comments received from readers of previous versions of this report, particularly from Phil Koopman (CMU), and from Brian Randell and Cliff Jones (Newcastle).

This material is partially based on work supported by the United States Air Force and DARPA under contract FA8750-23-C-0519. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Air Force and DARPA.

References

[1] *ASCAD: Adelard Safety Case Development Manual*. Adelard LLP, London, UK, 1998.

[2] Rohan Agarwal et al. Many-shot in-context learning. arXiv:2404.11018, April 2024.

[3] Erin E. Alves, Devesh Bhatt, Brendan Hall, Kevin Driscoll, Anitha Murugesan, and John Rushby. Considerations in assuring safety of increasingly autonomous systems. NASA Contractor Report NASA/CR-2018-220080, NASA Langley Research Center, July 2018.

[4] Christopher R. Anderson and Louise A. Dennis. Autonomous systems' safety cases for use in UK nuclear environments. arXiv:2310.02344, October 2023.

[5] Ian A. Apperly. What is "Theory of Mind"? Concepts, cognitive processes and individual differences. *Quarterly Journal of Experimental Psychology*, 65(5):825–839, 2012.

[6] Isaac Asimov. *I Robot*. Gnome Press, 1950.

[7] ASTM. *Standard Practice for Methods to Safely Bound Flight Behavior of Unmanned Aircraft Systems Containing Complex Functions*. ASTM (American Society for Testing and Materials), 2017. ASTM F3269-17.

[8] Angello Astorga, Chiao Hsieh, P. Madhusudan, and Sayan Mitra. Perception contracts for safety of ML-enabled systems. *Proceedings of the ACM on Programming Languages*, 7(OOPSLA2):2196–2223, October 2023.

[9] Bianca Baggiarini. Israel's AI can produce 100 bombing targets a day in Gaza. Is this the future of war? *The Conversation*, December 2023. https://theconversation.com/israels-ai-can-produce-100-bombing-targets-a-day-in-gaza

[10] Yuntao Bai et al. Constitutional AI: Harmlessness from AI feedback. arXiv:2212.08073, December 2022.

[11] Yuntao Bai et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. arXiv:2204.05862, April 2022.

[12] Lisanne Bainbridge. Ironies of automation. *Automatica*, 19(6):775–779, 1983.

[13] Clark Barrett, Pascal Fontaine, and Cesare Tinelli. The SMT-LIB Standard: Version 2.6. Technical report, Department of Computer Science, The University of Iowa, May 2021. Available at www.SMT-LIB.org.

[14] Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency (FAccT)*, pages 610–623, 2021.

[15] Ruth Benedict. *The Chrysanthemum and the Sword: Patterns of Japanese Culture*. Houghton Mifflin, 1946.

[16] Yoshua Bengio et al. Managing AI risks in an era of rapid progress. arXiv:2310.17688, October 2023.

[17] Yoshua Bengio et al. *International Scientific Report on the Safety of Advanced AI: Interim Report*. UK DSIT research paper series number 2024/009, May 2024. https://www.gov.uk/government/publications/international-scientific-report-on-the-safety-of-advanced-ai.

[18] Max Bennett. *A Brief History of Intelligence: Evolution, AI, and the Five Breakthroughs that Made Our Brains*. Mariner Books, 2023.

[19] Saddek Bensalem et al. What, indeed, is an achievable provable guarantee for learning-enabled safety critical systems. arXiv:2307.11784, July 2023.

[20] Derek Bickerton. *Adam's Tongue: How Humans Made Language, How Language Made Humans*. Macmillan, 2009.

[21] Peter Bishop. Does software have to be ultra reliable in safety critical systems? In SafeComp [152], pages 118–129.

[22] Peter Bishop, Andrey Povyakalo, and Lorenzo Strigini. Bootstrapping confidence in future safety from past safe operation. In *IEEE 33rd International Symposium on Software Reliability Engineering (ISSRE)*, pages 97–108, Charlotte, NC, October 2022.

[23] Robin Blades. AI generates hypotheses human scientists have not thought of. *Scientific American*, October 2021.

[24] Robin Bloomfield and Peter Bishop. Safety and assurance cases: Past, present and possible future—an Adelard perspective. In Chris Dale and Tom Anderson, editors, *Advances in System Safety: Proceedings of the Nineteenth Safety-Critical Systems Symposium*, pages 51–67, Springer, Bristol, UK, February 2010.

[25] Robin Bloomfield, Gareth Fletcher, Heidy Khlaaf, Luke Hinde, and Philippa Ryan. Safety case templates for autonomous systems. arXiv:2102.02625, 2021.

[26] Robin Bloomfield, Heidy Khlaaf, Philippa Ryan Conmy, and Gareth Fletcher. Disruptive innovations and disruptive assurance: Assuring machine learning and autonomy. *IEEE Computer*, 52(9):82–89, 2019.

[27] Robin Bloomfield, Kate Netkachova, and John Rushby. Defeaters and eliminative argumentation in Assurance 2.0. Technical report, Computer Science Laboratory, SRI International, Menlo Park, CA, May 2024. Also arXiv:2405.15800.

[28] Robin Bloomfield and John Rushby. Assurance 2.0: A Manifesto. In Mike Parsons and Mark Nicholson, editors, *Systems and Covid-19: Proceedings of the 29th Safety-Critical Systems Symposium (SSS'21)*, pages 85–108, Safety-Critical Systems Club, York, UK, February 2021. Preprint available as arXiv:2004.10474.

[29] Robin Bloomfield and John Rushby. Confidence in Assurance 2.0. Technical report, Computer Science Laboratory, SRI International, Menlo Park, CA, May 2022. Updated May 2024. Also available as arXiv:2205.04522.

[30] Carolyn Boettcher, Rance DeLong, John Rushby, and Wilmar Sifre. The MILS component integration approach to secure information sharing. In *27th AIAA/IEEE Digital Avionics Systems Conference*, The Institute of Electrical and Electronics Engineers, St. Paul, MN, October 2008. Best Paper award.

[31] Nick Bostrom. *Superintelligence: Paths, Dangers, Strategies*. Oxford University Press., 2014.

[32] Selmer Bringsjord, Konstantine Arkoudas, and Paul Bello. Toward a general logicist methodology for engineering ethically correct robots. *IEEE Intelligent Systems*, 21(4):38–44, 2006.

[33] Richard Brown, Hakwan Lau, and Joseph E. LeDoux. Understanding the higher-order approach to consciousness. *Trends in Cognitive Sciences*, 23(9):754–768, 2019.

[34] Tom Brown et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901, 2020.

[35] Tom B. Brown et al. Adversarial patch. arXiv:1712.09665, 2017.

[36] Lawrence D. Burns and Christopher Shulgan. *Autonomy: The Quest to Build the Driverless Car—And How It Will Reshape Our World*. Harper Collins, August 2018.

[37] Ricky W. Butler and George B. Finelli. The infeasibility of experimental quantification of life-critical software reliability. *IEEE Transactions on Software Engineering*, 19(1):3–12, January 1993.

[38] Sidney Carls-Diamante. Where is it like to be an octopus? *Frontiers in Systems Neuroscience*, 16, 2022.

[39] Esra Acar Celik et al. Application of STPA for the elicitation of safety requirements for a machine learning-based perception component in automotive. In Mario Trapp et al., editors, *Computer Safety, Reliability, and Security (*SafeComp *2022)*, Volume 13414 of Springer *Lecture Notes in Computer Science*, pages 319–332, Springer, Munich, Germany, September 2022.

[40] James Christie. The Post Office Horizon IT scandal and the presumption of the dependability of computer evidence. *Digital Evidence & Electronic Signature Law Review*, 17:49, 2020.

[41] Agata Ciabattoni et al. *Normative Reasoning for AI: Report from Dagstuhl Seminar 23151*, April 2023. Available from https://drops.dagstuhl.de/storage/04dagstuhl-reports/volume13/issue04/23151/DagRep.13.4.1/DagRep.13.4.1.pdf.

[42] Ilona Cieslik et al. State of the art study of the safety argumentation frameworks for automated driving system safety. arXiv:2302.00437, January 2023.

[43] E. M. Clarke, O. Grumberg, S. Jha, Y. Lu, and H. Veith. Counterexample-guided abstraction refinement. In E. A. Emerson and A. P. Sistla, editors, *Computer-Aided Verification, CAV '2000*, Volume 1855 of Springer-Verlag *Lecture Notes in Computer Science*, pages 154–169, Springer-Verlag, Chicago, IL, July 2000.

[44] Joshua Clymer, Nick Gabrieli, David Krueger, and Thomas Larsen. Safety cases: How to justify the safety of advanced AI systems. arXiv:2403.10462, March 2024.

[45] Roger C. Conant and W. Ross Ashby. Every good regulator of a system must be a model of that system. *International Journal of Systems Science*, 1(2):89–97, 1970.

[46] David Dalrymple et al. Towards guaranteed safe AI: A framework for ensuring robust and reliable AI systems. arXiv:2405.06624, June 2024.

[47] Sopam Dasgupta, Farhad Shakerin, Joaquín Arias, Elmer Salazar, and Gopal Gupta. Counterfactual explanation generation with s(CASP). arXiv:2310.14497, October 2023.

[48] Evan T. Dill, Steven D. Young, and Kelly J. Hayhurst. SAFEGUARD: An assured safety net technology for UAS. In *35th AIAA/IEEE Digital Avionics Systems Conference*. IEEE, 2016.

[49] Yi Dong et al. Reliability assessment and safety arguments for machine learning components in system assurance. *ACM Transactions on Embedded Computing Systems*, 22(3):1–48, 2023. Draft available as arXiv:2112.00646.

[50] Richard O. Duda and Peter E. Hart. Use of the Hough transformation to detect lines and curves in pictures. *Communications of the ACM*, 15(1):11–15, 1972.

[51] Kristina Dzeparoska, Jieyu Lin, Ali Tizghadam, and Alberto Leon-Garcia. LLM-based policy generation for intent-based management of applications. In *19th IEEE International Conference on Network and Service Management (CNSM)*, pages 1–7, 2023.

[52] Mary T. Dzindolet et al. The role of trust in automation reliance. *International Journal of Human-Computer Studies*, 58(6):697–718, 2003.

[53] Amnon H. Eden et al., editors. *Singularity hypotheses: A scientific and philosophical assessment*. Springer, The Frontiers Collection, 2012.

[54] David Edmonds. *Would you Kill the Fat Man? The Trolley Problem and What your Answer Tells us About Right and Wrong*. Princeton University Press, 2013.

[55] Daniel Everett. *How Language Began: The Story of Humanity's Greatest Invention*. Profile Books, 2017.

[56] FAA. *System Design and Analysis*. Federal Aviation Administration, June 21, 1988. Advisory Circular 25.1309-1A.

[57] Francesca Favaro et al. Building a credible case for safety: Waymo's approach for the determination of absence of unreasonable risk. arXiv:2306.01917, June 2023.

[58] Nick Feng et al. Analyzing and debugging normative requirements via satisfiability checking. arXiv:2401.05673, January 2024.

[59] Bruce A. Francis and Walter M. Wonham. The internal model principle of control theory. *Automatica*, 12(5):457–465, 1976.

[60] Keith Frankish. Dual-process and dual-system theories of reasoning. *Philosophy Compass*, 5(10):914–926, 2010.

[61] David Gamez. Progress in machine consciousness. *Consciousness and cognition*, 17(3):887–910, 2008.

[62] Michael S. Gazzaniga. *Who's in Charge?: Free Will and the Science of the Brain*. Harper Collins, 2012.

[63] Rocco J. Gennaro. *Higher-Order Theories of Consciousness: An Anthology*, volume 56 of *Advances in Consciousness Research*. John Benjamins Publishing, 2004.

[64] I. J. Good. Weight of evidence: A brief survey. In J.M Bernardo et al., editors, *Bayesian Statistics 2: Proceedings of the Second Valencia International Meeting*, pages 249–270, Valencia, Spain, September 1983.

[65] Naveen Sundar Govindarajulu and Selmer Bringsjord. On automating the doctrine of double effect. arXiv:1703.08922, 2017.

[66] Ryan Greenblatt, Buck Shlegeris, Kshitij Sachan, and Fabien Roger. AI control: Improving safety despite intentional subversion. arXiv:2312.06942, December 2023.

[67] Dylan Hadfield-Menell, Stuart J. Russell, Pieter Abbeel, and Anca Dragan. Cooperative inverse reinforcement learning. In *Advances in Neural Information Processing Systems*, Volume 29, 2016.

[68] Jonathan Haidt. *The Righteous Mind: Why Good People Are Divided by Politics and Religion*. Vintage, 2013. Paperback edition.

[69] Code of Hammurabi, c. 1772 BC. English translation by L.W. King available at http://eawc.evansville.edu/anthology/hammurabi.htm.

[70] C. A. J. Hanselaar et al. The safety shell: an architecture to handle functional insufficiencies in automated driving. arXiv:2311.08413, November 2024.

[71] Richard Hawkins and Philippa Ryan Conmy. Identifying run-time monitoring requirements for autonomous systems through the analysis of safety arguments. In *Computer Safety, Reliability, and Security (*SafeComp *2023)*, Volume 14181 of Springer *Lecture Notes in Computer Science*, pages 11–24, Springer, Toulouse, France, September 2023.

[72] Michael Townsen Hicks, James Humphries, and Joe Slater. ChatGPT is bullshit. *Ethics and Information Technology*, 26(2):38, 2024.

[73] Donald Hoffman. *The Case Against Reality: Why Evolution Hid the Truth from Our Eyes*. WW Norton & Company, 2019.

[74] Owen Holland and Rod Goodman. Robots with internal models: A route to machine consciousness? *Journal of Consciousness Studies*, 10(4-5):77–109, 2003.

[75] C. Michael Holloway. Understanding the Overarching Properties. Technical Memorandum NASA/TM-2019-220292, NASA Langley Research Center, Hampton VA, July 2019.

[76] Boyue Caroline Hu et al. Towards requirements specification for machine-learned perception based on human performance. In *2020 IEEE Seventh International Workshop on Artificial Intelligence for Requirements Engineering (AIRE)*, pages 48–51, IEEE, Zurich, Switzerland, September 2020.

[77] Xiaowei Huang et al. A survey of safety and trustworthiness of deep neural networks: Verification, testing, adversarial attack and defence, and interpretability. *Computer Science Review*, 37:100270, 2020. Also available as arXiv:1812.08342.

[78] Evan Hubinger et al. Sleeper Agents: Training deceptive LLMs that persist through safety training. arXiv:2401.05566, January 2024.

[79] Geoffrey Irving and Amanda Askell. AI safety needs social scientists. *Distill*, February 2019. DOI:10.23915/distill.00014.

[80] Takuto Ishimatsu et al. Modeling and hazard analysis using STPA. In *Proceedings of the 4th IAASS Conference, Making Safety Matter*, Huntsville, AL, May 2010. https://dspace.mit.edu/handle/1721.1/79639.

[81] Daniel Jackson, Martyn Thomas, and Lynette I. Millett, editors. *Software for Dependable Systems: Sufficient Evidence?* National Academies Press, Washington, DC, May 2007.

[82] Michael Jackson. *Software requirements & specifications: A Lexicon of Practice, Principles and Prejudices*. ACM Press/Addison-Wesley Publishing Co., 1995.

[83] Alon Jacovi, Ana Marasović, Tim Miller, and Yoav Goldberg. Formalizing trust in artificial intelligence: Prerequisites, causes and goals of human trust in AI. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency (FAccT)*, pages 624–635, Held virtually, Canada, March 2021.

[84] Sumit Kumar Jha, Susmit Jha, Patrick Lincoln, Nathaniel D. Bastian, Alvaro Velasquez, Rickard Ewetz, and Sandeep Neema. Counterexample guided inductive synthesis using large language models and satisfiability solving.

In *IEEE Military Communications Conference (MILCOM)*, pages 944–949, IEEE, Boston, MA, October 2023.

[85] Susmit Jha, John Rushby, and N. Shankar. Model-centered assurance for autonomous systems. In António Casimiro et al., editors, *Computer Safety, Reliability, and Security (SafeComp 2020)*, Volume 12234 of Springer *Lecture Notes in Computer Science*, pages 228–243, Springer, Lisbon, Portugal, September 2020.

[86] Rolf Johansson and Philip Koopman. Continuous learning approach to safety engineering. In *CARS—Critical Automotive Applications: Robustness & Safety*, Zaragoza, Spain, September 2022. https://hal.science/CARS2022/hal-03782627v1.

[87] Cheng Ju, Aurélien Bibaut, and Mark van der Laan. The relative performance of ensemble methods with deep convolutional neural networks for image classification. *Journal of Applied Statistics*, 45(15):2800–2818, 2018.

[88] Leslie P. Kaelbling, Michael L. Littman, and Andrew W. Moore. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4:237–285, May 1996.

[89] Daniel Kahneman. *Thinking, Fast and Slow*. Farrar, Straus and Giroux, 2011.

[90] Nidhi Kalra and Susan M. Paddock. Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability? *Transportation Research Part A: Policy and Practice*, 94:182–193, 2016.

[91] Tim Kelly. *Arguing Safety—A Systematic Approach to Safety Case Management*. DPhil thesis, Department of Computer Science, University of York, UK, 1998.

[92] Heidy Khlaaf. *Toward Comprehensive Risk Assessments and Assurance of AI-Based Systems*. Trail of Bits, 2023.

[93] Diederik P. Kingma and Max Welling. Auto-encoding variational Bayes. arXiv:1312.6114, 2013.

[94] Robert Kirk. *Zombies*. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Spring 2019 edition.

[95] Alistair Knott et al. Generative AI models should include detection mechanisms as a condition for public release. *Ethics and Information Technology*, 25(4):55, October 2023.

[96] Nicole Kobie. The complicated truth about China's social credit system. Wired UK, January 2019. https://www.wired.co.uk/article/china-social-credit-system-explained.

[97] Phil Koopman. Anatomy of a robotaxi crash: The Cruise pedestrian dragging mishap. In *Computer Safety, Reliability, and Security (*SafeComp *2024)*, Lecture Notes in Computer Science, Springer, Florence, Italy, September 2024. To appear.

[98] Philip Koopman. The heavy tail safety ceiling. In *SAE Automated and Connected Vehicle Systems Testing Symposium*, Volume 1145, pages 8950–8961, Greenville, SC, June 2018.

[99] Philip Koopman, Beth Osyk, and Jack Weast. Autonomous vehicles meet the physical world: RSS, variability, uncertainty, and proving safety. In *Computer Safety, Reliability, and Security (*SafeComp *2019)*, Volume 11698 of Springer *Lecture Notes in Computer Science*, pages 245–253, Springer, Turku, Finland, September 2019.

[100] Hermann Kopetz. *An Architecture for Safe Driving Automation*. In *Principles of Systems Design: Essays Dedicated to Thomas A. Henzinger on the Occasion of His 60th Birthday*, pages 61–84. Springer, 2022.

[101] Hermann Kopetz et al. *Emergence in Cyber-Physical Systems-of-Systems*. In Andrea Bondavalli, Sara Bouchenak, and Hermann Kopetz, editors, *Cyber-Physical Systems of Systems*, volume 10099 of *Lecture Notes in Computer Science*, pages 73–96. Springer-Verlag, 2016.

[102] Hermann Kopetz and Wilfried Steiner. *Real-Time Systems: Design Principles for Distributed Embedded Applications*. Springer, 2022.

[103] Piotr Kulicki, Robert Trypuz, and Michael P. Musielewicz. Towards a formal ethics for autonomous cars. In *14th International Conference on Deontic Logic and Normative Systems (DEON)*, Utrecht, The Netherlands, July 2018.

[104] Ram Shankar Siva Kumar et al. Adversarial machine learning—industry perspectives. arXiv:2002.05646, February 2020.

[105] J. C. Laprie, editor. *Dependability: Basic Concepts and Terminology in English, French, German, Italian and Japanese*, Volume 5 of Springer-Verlag, Vienna, Austria *Dependable Computing and Fault-Tolerant Systems*. Springer-Verlag, Vienna, Austria, February 1991.

[106] Seth Lazar. Frontier AI ethics: Anticipating and evaluating the societal impacts of generative agents. arXiv:2404.06750, April 2024.

[107] Peter R. Lewis and Stephen Marsh. What is it like to trust a rock? a functionalist perspective on trust and trustworthiness in artificial intelligence. *Cognitive Systems Research*, 72:33–49, 2022.

[108] Bei Li et al. Deliberate then generate: Enhanced prompting framework for text generation. arXiv:2305.19835, May 2023.

[109] Benjamin Libet. Unconscious cerebral initiative and the role of conscious will in voluntary action. *Behavioral and Brain Sciences*, 8(4):529–539, 1985.

[110] Felix Lindner and Martin Mose Bentzen. A formalization of Kant's second formulation of the Categorical Imperative. In *14th International Conference on Deontic Logic and Normative Systems (DEON)*, Utrecht, The Netherlands, July 2018.

[111] B. Littlewood and D. R. Miller. Conceptual modeling of coincident failures in multiversion software. *IEEE Transactions on Software Engineering*, 15(12):1596–1614, December 1989.

[112] Bev Littlewood and John Rushby. Reasoning about the reliability of diverse two-channel systems in which one channel is "possibly perfect". *IEEE Transactions on Software Engineering*, 38(5):1178–1194, September/October 2012.

[113] Bev Littlewood and Lorenzo Strigini. Validation of ultrahigh dependability for software-based systems. *Communications of the ACM*, pages 69–80, November 1993.

[114] Yixin Liu et al. Sora: A review on background, technology, limitations, and opportunities of large vision models. arXiv:2402.17177, February 2024.

[115] Ravi Mangal et al. Concept-based analysis of neural networks via vision-language models. arXiv:2403.19837, March 2024.

[116] Nahema Marchal et al. Generative AI misuse: A taxonomy of tactics and insights from real-world data. arXiv:2406.13843, June 2024.

[117] Paul Marshall et al. Recommendations for the probity of computer evidence. *Digital Evidence & Electronic Signature Law Review*, 18:18, 2021.

[118] Jennifer Mather. What is in an octopus's mind? *Animal Sentience*, 4(26):1, 2019.

[119] J. A. McDermid, M. Nicholson, D. J. Pumfrey, and P. Fenelon. Experience with the application of HAZOP to computer-based systems. In *COMPASS '96 (Proceedings of the Eleventh Annual Conference on Computer Assurance)*, pages 37–48, IEEE Washington Section, Gaithersburg, MD, June 1996.

[120] Alain Mebsout and Cesare Tinelli. Proof certificates for SMT-based model checkers for infinite-state systems. In *2016 Formal Methods in Computer-Aided Design (FMCAD)*. pages 117–124, IEEE, 2016.

[121] Ayhan Mehmed, Moritz Antlanger, and Wilfried Steiner. The monitor as key architecture element for safe self-driving cars. In *50th Annual IEEE-IFIP International Conference on Dependable Systems and Networks-Supplemental Volume (DSN-S)*, pages 9–12, IEEE, Valencia, Spain, June 2020.

[122] Ayhan Mehmed et al. Early concept evaluation of a runtime monitoring approach for safe automated driving. In *IEEE Zooming Innovation in Consumer Technologies Conference (ZINC)*, pages 53–58, May 2022.

[123] Ayhan Mehmed, Wilfried Steiner, and Aida Čaušević. Formal verification of an approach for systematic false positive mitigation in safe automated driving system, 2020. Available at http://www.es.mdh.se/publications/5794-.

[124] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1765–1773, 2017.

[125] *Vehicle Automation Report; Tempe, AZ*. National Transportation Safety Board, November 2019. HWY18MH010.

[126] David Nistér, Hon-Leung Lee, Julia Ng, and Yizhou Wang. The safety force field. NVIDIA White Paper, March 2019. Available at https://www.nvidia.com/en-us/self-driving-cars/safety-force-field/.

[127] OpenAI. Prompt engineering. https://platform.openai.com/docs/guides/prompt-engineering, 2023.

[128] Long Ouyang et al. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, Volume 35, pages 27730–27744, 2022. Also available as arXiv:2203.02155.

[129] Charles Perrow. *Normal Accidents: Living with High Risk Technologies*. Basic Books, New York, NY, 1984.

[130] Arianna Pipitone, Francesco Lanza, Valeria Seidita, and Antonio Chella. Inner speech for a self-conscious robot. In Antonio Chella et al., editors, *Towards Conscious AI Systems Symposium (TOCAIS): AAAI Spring Symposium Series*, Stanford, CA, March 2019.

[131] Maciej P. Polak and Dane Morgan. Extracting accurate materials data from research papers with conversational language models and prompt engineering. *Nature Communications*, 15(1):1569, 2024.

[132] Karl Popper. Three worlds. The Tanner Lecture on Human Values, delivered at the University of Michigan, April 1978. Available at https://tannerlectures.utah.edu/_resources/documents/a-to-z/p/popper80.pdf.

[133] Karl R. Popper and John C. Eccles. *The Worlds 1, 2 and 3*. In *The Self and Its Brain*, pages 36–50. Springer, 1977.

[134] Xiangyu Qi et al. Fine-tuning aligned language models compromises safety, even when users do not intend to! arXiv:2310.03693, October 2023.

[135] Prabhakar Raghavan. Gemini image generation got it wrong. we'll do better. Google blog post, February 2024. https://blog.google/products/gemini/gemini-image-generation-issue/.

[136] Abhiramon Rajasekharan, Yankai Zeng, Parth Padalkar, and Gopal Gupta. Reliable natural language understanding with large language models and answer set programming. arXiv:2302.03780, February 2023.

[137] Brian Randell and Brian Coghlan. ChatGPT's astonishing fabrications about Percy Ludgate. *IEEE Annals of the History of Computing*, 45(2):71–72, 2023.

[138] John Rawls. *A Theory of Justice*. Belknap Press/Harvard University Press, 1971.

[139] Vipula Rawte, Amit Sheth, and Amitava Das. A survey of hallucination in large foundation models. arXiv:2309.05922, September 2023.

[140] James A. Reggia. The rise of machine consciousness: Studying consciousness with computational models. *Neural Networks*, 44:112–131, 2013.

[141] Bernadino Romera-Paredes et al. Mathematical discoveries from program search with large language models. *Nature*, December 2023.

[142] *Runtime Verification home page*. http://www.runtime-verification.org/.

[143] John Rushby. Quality measures and assurance for AI software. Technical Report SRI-CSL-88-7R, Computer Science Laboratory, SRI International, Menlo Park, CA, September 1988. Also available as NASA Contractor Report 4187.

[144] John Rushby. Partitioning for avionics architectures: Requirements, mechanisms, and assurance. NASA Contractor Report CR-1999-209347, NASA Langley Research Center, June 1999. Available at https://www.csl.sri.com/~rushby/abstracts/partitioning.

[145] John Rushby. Bus architectures for safety-critical embedded systems. In Tom Henzinger and Christoph Kirsch, editors, *EMSOFT 2001: Proceedings of the First Workshop on Embedded Software*, Volume 2211 of Springer-Verlag *Lecture Notes in Computer Science*, pages 306–323, Springer-Verlag, Lake Tahoe, CA, October 2001.

[146] John Rushby. Runtime certification. In Martin Leucker, editor, *Eighth Workshop on Runtime Verification: RV08*, Volume 5289 of Springer-Verlag *Lecture Notes in Computer Science*, pages 21–35, Springer-Verlag, Budapest, Hungary, April 2008.

[147] John Rushby. The interpretation and evaluation of assurance cases. Technical Report SRI-CSL-15-01, Computer Science Laboratory, SRI International, Menlo Park, CA, July 2015. Available at http://www.csl.sri.com/users/rushby/papers/sri-csl-15-1-assurance-cases.pdf.

[148] John Rushby. The indefeasibility criterion for assurance cases. In Yamine Ait-Ameur, Shin Nakajima, and Dominique Méry, editors, *Implicit and Explicit Semantics Integration in Proof Based Developments of Discrete Systems*, Communications of NII Shonan Meetings, pages 259–279, Springer, Kanagawa, Japan, July 2020. Postproceedings of a workshop held in November 2016.

[149] John Rushby. On computational mechanisms for shared intentionality and speculation on rationality and consciousness. Technical report, Computer Science Laboratory, SRI International, Menlo Park, CA, June 2023. Also available as arXiv:2306.13657.

[150] John Rushby and Daniel Sanchez. Technology and consciousness. Technical report, Computer Science Laboratory, SRI International, Menlo Park, CA, September 2018. Minor update available as arXiv:2209.03956.

[151] Stuart Russell. *Human-Compatible Artificial Intelligence.* In Stephen Muggleton and Nick Chater, editors, *Human-Like Machine Intelligence*, chapter 1, pages 3–23. Oxford University Press, 2022. Preprint available at https://aima.eecs.berkeley.edu/~russell/papers/mi19book-hcai.pdf.

[152] SafeComp *2013: Proceedings of the 32nd International Conference on Computer Safety, Reliability, and Security*, Volume 8153 of Springer-Verlag *Lec-*

ture Notes in Computer Science, Toulouse, France, September 2013. Springer-Verlag.

[153] Rick Salay and Krzysztof Czarnecki. A safety assurable human-inspired perception architecture. In *Computer Safety, Reliability, and Security. SAFECOMP 2022 Workshops: DECSoS, DepDevOps, SASSUR, SENSEI, USDAI, and WAISE Munich, Germany, September 6–9, 2022, Proceedings*. pages 302–315, Springer, 2022. Also arXiv:2205.07862.

[154] Rick Salay et al. The missing link: Developing a safety case for perception components in automated driving. arXiv:2108.13294, September 2022.

[155] Bruce Schneier. AI and trust. https://www.belfercenter.org/publication/ai-and-trust, November 2023.

[156] Congressional Research Service. Defense primer: U.S. policy on lethal autonomous weapon systems. https://crsreports.congress.gov/product/pdf/IF/IF11150, February 2024.

[157] Ilia Shumailov et al. AI models collapse when trained on recursively generated data. *Nature*, 631(8022):755–759, July 2024.

[158] Linda J. Skitka, Kathleen L. Mosier, and Mark Burdick. Does automation bias decision-making? *International Journal of Human-Computer Studies*, 51(5):991–1006, 1999.

[159] Dylan Slack et al. Counterfactual explanations can be manipulated. In *Advances in Neural Information Processing Systems*, Volume 34, pages 62–75, 2021.

[160] Mark Staples. Critical rationalism and engineering: Ontology. *Synthese*, 191(10):2255–2279, July 2014.

[161] Ingo Steinwart and Andreas Christmann. *Support Vector Machines*. Springer Science & Business Media, 2008.

[162] James W. A. Strachan et al. Testing theory of mind in large language models and humans. *Nature Human Behavior*, May 2024.

[163] Lorenzo Strigini and Andrey Povyakalo. Software fault-freeness and reliability predictions. In SafeComp [152], pages 106–117.

[164] Yang Sun, Christopher M. Poskitt, Xiaodong Zhang, and Jun Sun. REDriver: Runtime enforcement for autonomous vehicles. arXiv:2401.02253, January 2024. To be presented at ICSE '24.

[165] Christian Szegedy et al. Intriguing properties of neural networks. arXiv:1312.6199, 2013.

[166] Adly Templeton et al. Scaling monosemanticity: Extracting interpretable features from Claude 3 Sonnet. Anthropic, May 2024. https://transformer-circuits.pub/2024/scaling-monosemanticity/index.html.

[167] Michael Tomasello. *Origins of Human Communication*. MIT Press, 2010.

[168] Michael Tomasello and Malinda Carpenter. Shared intentionality. *Developmental Science*, 10(1):121–125, 2007.

[169] Greta Tuckute, Nancy Kanwisher, and Evelina Fedorenko. *Language in Brains, Minds, and Machines*. In *Annual Review of Neuroscience*, volume 47. April 2024.

[170] Shannon Vallor. *Technology and the Virtues: A Philosophical Guide to a Future Worth Wanting*. Oxford University Press, 2016.

[171] Srivatsan Varadarajan et al. Clarissa: Foundations, tools and automation for assurance cases. In *42nd AIAA/IEEE Digital Avionics Systems Conference*, Barcelona, Spain, October 2023.

[172] Apostol Vassilev, Alina Oprea, Alie Fordyce, and Hyrum Anderson. Adversarial machine learning: A taxonomy and terminology of attacks and mitigations. Technical Report NIST AI 100-2e2023, National Institute of Standards and Technology, January 2024. Available from https://doi.org/10.6028/NIST.AI.100-2e2023.

[173] Sahil Verma et al. Counterfactual explanations and algorithmic recourses for machine learning: A review. arXiv:2010.10596, October 2020.

[174] Anne von der Lieth Gardner. *An Artificial Intelligence Approach to Legal Reasoning*. MIT Press, 1987.

[175] Joseph Weizenbaum. ELIZA——a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1):36–45, 1966.

[176] Wanja Wiese and Thomas K. Metzinger. *Vanilla PP for Philosophers: A Primer on Predictive Processing*. In Thomas K. Metzinger and Wanja Wiese, editors, *Philosophy and Predictive Processing*, chapter 1. MIND Group, Frankfurt am Main, 2017.

[177] Wikipedia contributors. *AI Alignment*. https://en.wikipedia.org/wiki/AI_alignment.

[178] Wikipedia contributors. *Tay (chatbot)*. https://en.wikipedia.org/wiki/Tay_(chatbot).

[179] Ziwei Xu, Sanjay Jain, and Mohan Kankanhalli. Hallucination is inevitable: An innate limitation of large language models. arXiv:2401.11817, January 2024.

[180] Han Yu, Zhiqi Shen, Chunyan Miao, Cyril Leung, Victor R. Lesser, and Qiang Yang. Building ethics into artificial intelligence. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18)*, pages 5527–5533, Stockholm, Sweden, July 2018.

[181] Chiyuan Zhang et al. Understanding deep learning requires rethinking generalization. arxiv:1611.03530, February 2017.

[182] Xingyu Zhao et al. Assessing safety-critical systems from operational testing: A study on autonomous vehicles. *Information and Software Technology*, 128:106393, December 2020.

[183] Xingyu Zhao, Bev Littlewood, Andrey Povyakalo, Lorenzo Strigini, and David Wright. Modeling the probability of failure on demand (pfd) of a 1-out-of-2 system in which one channel is "quasi-perfect". In *Reliability Engineering and System Safety*. pages 230–245, Elsevier Ltd, February 2017.