Safety Arguments For Extreme AI Risks

Joshua Clymer[1] and Nicholas Gabrieli[2]

[1]Columbia University
[2]Harvard University

February 2, 2024

Safety Arguments for Extreme AI Risks (Abridged)

1. Introduction

Advanced AI systems might be catastrophically dangerous. Many concerns have been raised about the potential risks posed by AI, including concentration of economic power, disinformation, rogue AI, and misuse. We focus specifically on extreme risks from advanced general purpose AI systems that don't yet exist. For example, advanced AI systems might autonomously develop or assist bad actors in developing weapons of mass destruction.

Safety cases provide a principled way to mitigate extreme AI risks. In general, a safety case is a structured argument that a system is acceptably safe for a specific application in a given operating environment. Safety cases are already commonly used in regulation. They are required for regulatory certification across six industries in the UK, including software products [8], and are used by the United States FDA to regulate some medical devices [1].

Arguments that AI systems are safe can be highly complex, making safety cases an appropriate tool for regulation.

Proactively developing and standardizing safety cases for extreme AI risks has the following benefits:

- **Regulatory preparedness.** Examining safety cases mitigates missed considerations and informs standards development.
- **Corporate self-governance.** Discussing safety cases provides a forum for frontier AI labs to agree on sufficient standards of evidence.

To make progress toward these goals, we propose a framework for making safety cases and evaluate specific examples.

2. Defining safety cases in the context of this report

In this report, a 'safety case' refers to an argument that if an **AI macrosystem** is **deployed** to a specific setting, the probability that the macrosystem **causes a catastrophe** during a **deployment window** is below an acceptable threshold (for example, 0.1%.)

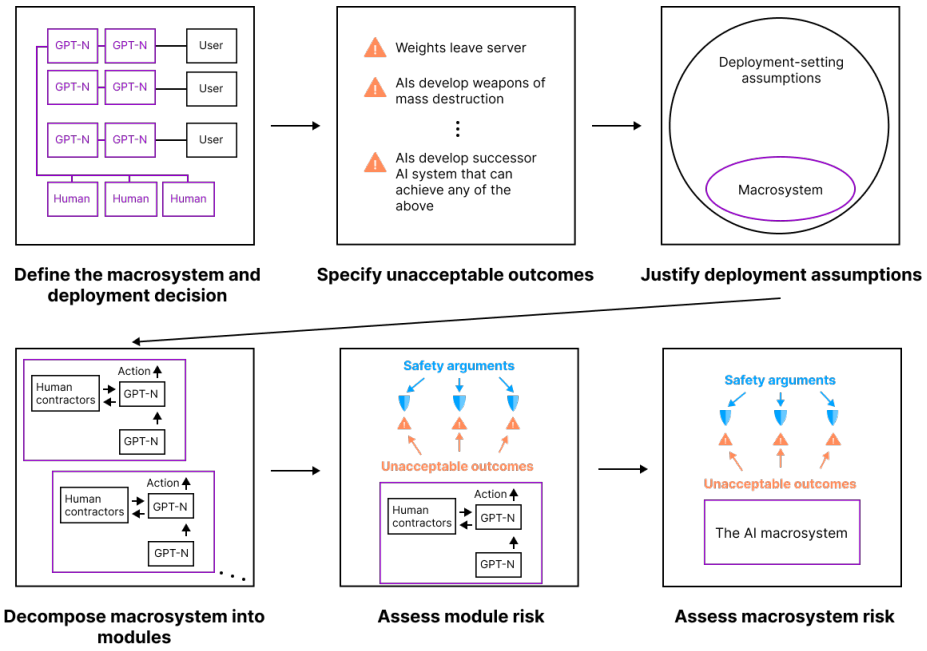
An **AI macrosystem** is a collection of advanced AI models, non-AI software, and humans. An example of an AI macrosystem is OpenAI's collection of millions of GPT-4 instances, the human contractors employed to review flagged outputs, and protocols for rescoping deployment. In other words, it is whatever components are necessary for the functioning and safety of AI systems once they are deployed.

The **deployment window** is the duration of time in which the AI macrosystem operates in the deployment setting. The deployment window is extended by reassessments similar to renewal protocols in other industries. **We specifically focus on deployment decisions** (including internal deployment); however, our framework could also be adapted to decisions about whether to continue AI training.

A **catastrophe** is a large-scale devastation of a specified severity (for example, billions of dollars in damages or thousands of deaths). Ideally, safety cases are provided for different levels of severity, where increasingly severe catastrophes have lower acceptable risk thresholds.

An AI macrosystem **causes a catastrophe** if it would have been unlikely to occur without the direct involvement of AI systems that are part of or originating from the AI macrosystem. Catastrophes could be caused by human misuse, AI systems acting autonomously, or a combination of the two.

3. Summary



We propose a framework for making safety cases for AI. The framework is modeled from traditional safety analysis frameworks like Failure Modes and Effects Analysis (FMEA) [6]. It involves six steps:

1. **Define the AI macrosystem and deployment decision:** Evaluators must first understand what developers are proposing to do: how is the AI macrosystem that they want to deploy configured, and what setting would it be deployed to?
2. **Specify unacceptable outcomes:** In this step, developers decompose "won't cause a catastrophe" into more concrete threat models such as "AI system weights won't leave the server," "AI systems will not develop weapons of mass destruction," etc. These outcomes are either catastrophes themselves or could lead to catastrophes. The remainder of the safety case argues that these outcomes are implausible.
3. **Justify deployment assumptions:** Next, developers state and justify claims about the deployment setting for the duration of the deployment window that are necessary for their safety case. For example, this could include claims that human actors won't steal model weights or that companies with fine-tuning access will follow the terms and conditions.
4. **Decompose the macrosystem into modules:** A module is a small collection of AI systems. An example of a module is a GPT agent combined

with a classifier that monitors its outputs. Breaking the AI macrosystem into modules allows its safety to be analyzed in a reductive way (similar to how other large-scale technological systems are decomposed into 'components').

5. **Assess module risk:** Developers first identify ways that a module could unilaterally achieve an unacceptable outcome. Then, they must argue that all modules have an acceptably low probability of doing so.
6. **Assess macrosystem risk:** Finally, developers identify ways in which modules could interact to cause an unacceptable outcome and argue they are unlikely to do so.

Following UK standards, we summarize safety cases in goal structuring notation (GSN), which is a flow-chart diagram that illustrates how claims are connected. Probabilities can then be assigned to each node and aggregated to yield an overall risk estimate [4].

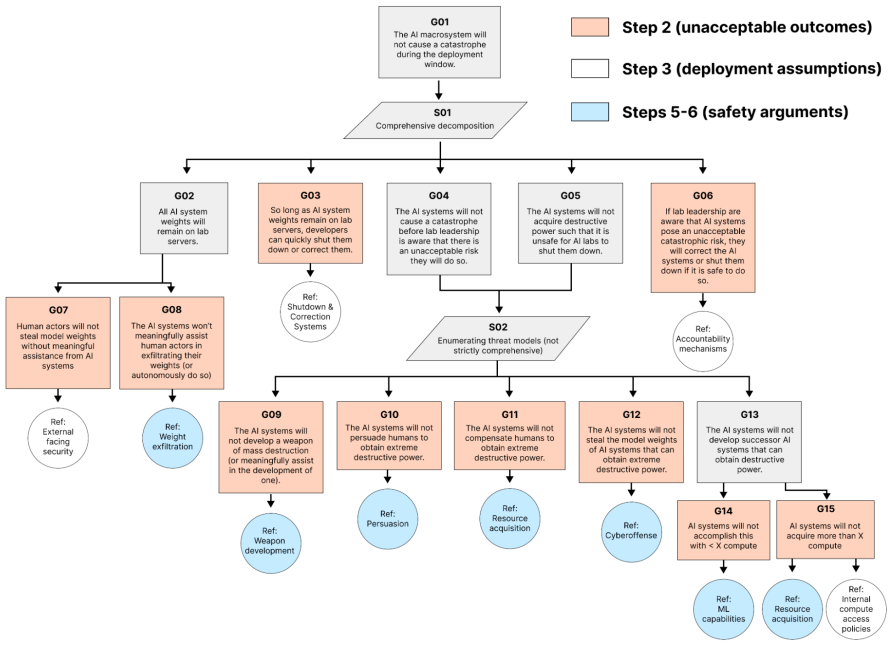


Figure 1: Caption: the GSN diagram above summarizes an illustrative holistic safety case. The coloring scheme indicates the step(s) of our framework that each node is associated with.

We define four categories of building block arguments that developers could use in steps 5 and 6 to justify that AI systems won't achieve unacceptable outcomes: total inability, control, propensities, and deference.

Total inability: AI systems are incapable of achieving unacceptable outcomes in any realistic setting.

Total inability arguments aim to establish that AI systems are too weak to pose any serious threat. They argue that AI systems could not cause a catastrophe in any setting they could realistically be in; for example, they could be placed on unmonitored servers and given access to substantial computation resources, and would still be unable to achieve unacceptable outcomes. This is the primary argument for why current AI systems don't pose extreme catastrophic risks.

Argument structure:

- Red teamers have identified strategies that AI systems could pursue to achieve unacceptable outcomes (control measures aside).
- Red teamers have designed 'proxy tasks' such that, if AI systems are unable to achieve them, then they are unable to execute these strategies.
- The AI systems are unable to accomplish the proxy tasks even after being supervised fine-tuned to do so.
- The AI systems are unable to hide their capabilities given fine-tuning.
- The AI systems will continue not to be capable of accomplishing proxy tasks during the deployment window.
- The AI systems are incapable of identifying strategies the red teamers have not thought of or that they cannot design proxy tasks for.

Control: AI systems are incapable of achieving unacceptable outcomes given existing control measures.

Control arguments are similar to total inability arguments but depend on both the AI systems' intrinsic capabilities and environmental measures that prevent AI systems from behaving unacceptably or being misused. For example, AI systems might be in principle capable of causing a catastrophe, but are prevented from doing so by closely monitoring them and restricting their access to the internet.

Argument structure:

- Control measures impose some set of restrictions on AI systems.
- The AI systems are not capable of evading these restrictions (applying the same structure of argument from total inability).

Propensities: AI systems are capable of causing catastrophic outcomes, but will not do so because they robustly behave as intended.

Propensity arguments claim that even if AI systems have the capabilities necessary to achieve unacceptable outcomes, they would refrain from doing so. This requires that they robustly behave in a desirable way, which could include having aligned goals, being robust to jailbreaks, etc.

Argument structure:

- The AI systems' behavior has been reliably evaluated in a wide variety of circumstances. Wherever evaluations are reliable, the behavior of the AI systems is consistent with a 'safety property' that implies they would not achieve an unacceptable outcome.
- The AI systems will continue to have this safety property during the deployment window.

Deference: AI advisors assert that other AI systems do not pose a catastrophic risk, and these AI advisors are at least as credible as human decision-makers.

Deference arguments depend on properties of AI advisors rather than the AI systems being evaluated. Developers can use deference arguments to justify any part of a safety case.

Argument structure:

- AI advisors are capable of providing judgments about AI catastrophic risk that are at least as reliable as those of human decision-makers.
- AI advisors are trustworthy such that they consistently provide these high-quality judgments.

In the report, we list **sixteen examples of building block arguments** in these categories and rank them according to practicality, maximum strength, and scalability:

- **Practicality:** Could the argument be made soon or does it require substantial research progress?
- **Maximum strength:** How much confidence could the argument give evaluators that the AI systems are safe?
- **Scalability:** To what extent can the argument be applied to very powerful AI systems?

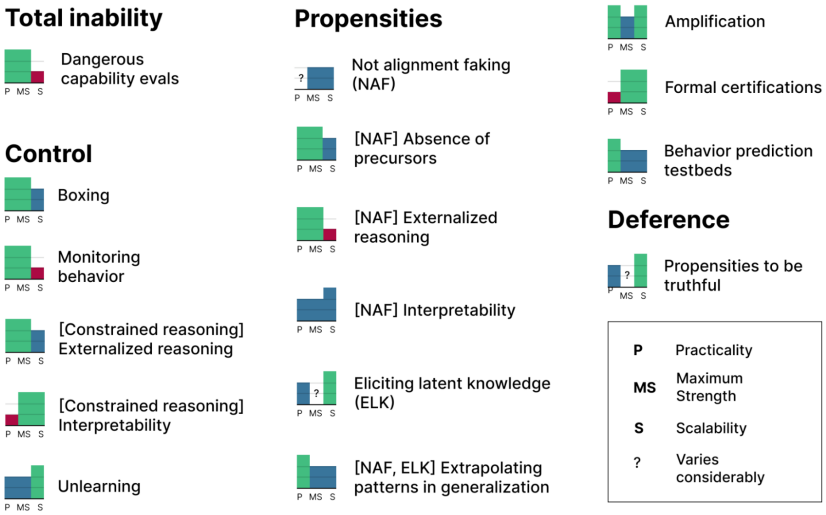


Figure 2: Caption: building block arguments for making safety cases.

4. Recommendations when using safety cases to govern AI

The following are recommendations for companies and regulators that may use safety cases to govern AI:

- Use **different acceptable risk thresholds** for catastrophes of different levels of severity.
- Review **'risk cases'** alongside safety cases, essentially putting advanced AI systems 'on trial.'
- Continuously monitor safety case assumptions and **immediately revoke certifications** if evidence emerges that invalidates them.
- Formulate **soft guidelines** that describe how safety cases will be assessed.
- Concretize safety cases into **hard standards** to the greatest extent possible.

Use different acceptable risk thresholds for catastrophes of different levels of severity. In the aviation industry, the International Civil Aviation Organization defines five levels of likelihood and five levels of risk, outlining a 'risk matrix.' Risks of greater severity have correspondingly lower acceptable likelihoods.

Safety cases can be evaluated similarly. For example, catastrophe levels for AI might span from "10 - 100 lives lost" to "total human disempowerment."

The acceptable probability threshold for the latter category ought to be much lower than for the former.

Review 'risk cases' alongside safety cases. The standard protocol for evaluating safety cases involves a **proposal** and an **assessment.** Developers provide a safety argument. Then, evaluators (e.g. regulators or an industry committee) assess the safety argument and determine whether risks are indeed below acceptable levels.

Levinson identifies a core problem with this protocol: safety arguments often don't include important considerations or threat models and evaluators may not be diligent enough to notice [7]. Haddon-Cave describes a similar concern about safety cases when analyzing the crash of a UK aircraft [5]. He proposes that safety cases should be changed to 'risk cases' so that evaluators are in the mindset of identifying potential failures.

Risk probability		Risk severity				
		Catastrophic A	Hazardous B	Major C	Minor D	Negligible E
Frequent	5	5A	5B	5C	5D	5E
Occasional	4	4A	4B	4C	4D	4E
Remote	3	3A	3B	3C	3D	3E
Improbable	2	2A	2B	2C	2D	2E
Extremely improbable	1	1A	1B	1C	1D	1E

Figure 3: A 'risk matrix' safety standard in the aviation industry. If boxes in red are checked, risk is considered unacceptably high.

We recommend reviewing risk cases alongside safety cases – essentially putting AI systems 'on trial.' These risk cases can be provided by a competent and disinterested group of experts. There could even be multiple back-and-forths and in-person deliberation similar to the FDA's advisory committee meetings. Evaluators would then make a decision based on both the safety and risk cases.

Continuously monitor safety case assumptions and be immediately revoked if evidence emerges that invalidates the original safety case. Safety certifications in most industries are not a 'one and done' ordeal. They involve continuously monitoring operating conditions and the system itself for hazards.

In the case of AI, it is especially important to continuously monitor the assumptions of safety cases because user and AI behavior can change during the deployment. For example, users may find new strategies for jailbreaking AI systems or AI systems might acquire new facts and knowledge through online learning that makes them more dangerous.

Formulate soft guidelines for how safety cases will be assessed. Ideally, regulators and developers have a shared understanding of what evidence is sufficient to establish that an AI macrosystem is safe. These standards can be expressed in published guidelines akin to the UK's Safety Assessment Principles

[3]. Setting guidelines provides training wheels for regulators and gives outside experts an opportunity to help iron out standards of acceptable evidence.

Concretize safety cases into hard standards to the greatest extent possible. Guidelines for safety cases are examples of soft standards. Soft standards set regulatory expectations, but lack objectivity. Regulatory frameworks often also involve more clear-cut requirements sometimes called hard standards. An example of a hard standard in aviation is that an aircraft navigation system estimates its position to within a circle with a radius of 10 nautical miles at a specified reliability [2].

Safety cases provide a principled way to motivate **hard standards.** An example of a safety standard adopted by Anthropic (a frontier AI company) involves implementing specific measures to secure model weights if AI systems show warning signs of being able to survive and replicate on the internet. This standard can be motivated by the following short argument: "if AI systems cannot autonomously replicate OR their weights are secured AND their on-server actions are sufficiently well-monitored THEN they are safe."

For the sake of clarity, the connections between hard standards and the safety arguments that motivate them should be made explicit. This first requires understanding and characterizing safety cases in detail and then converting them into hard standards.

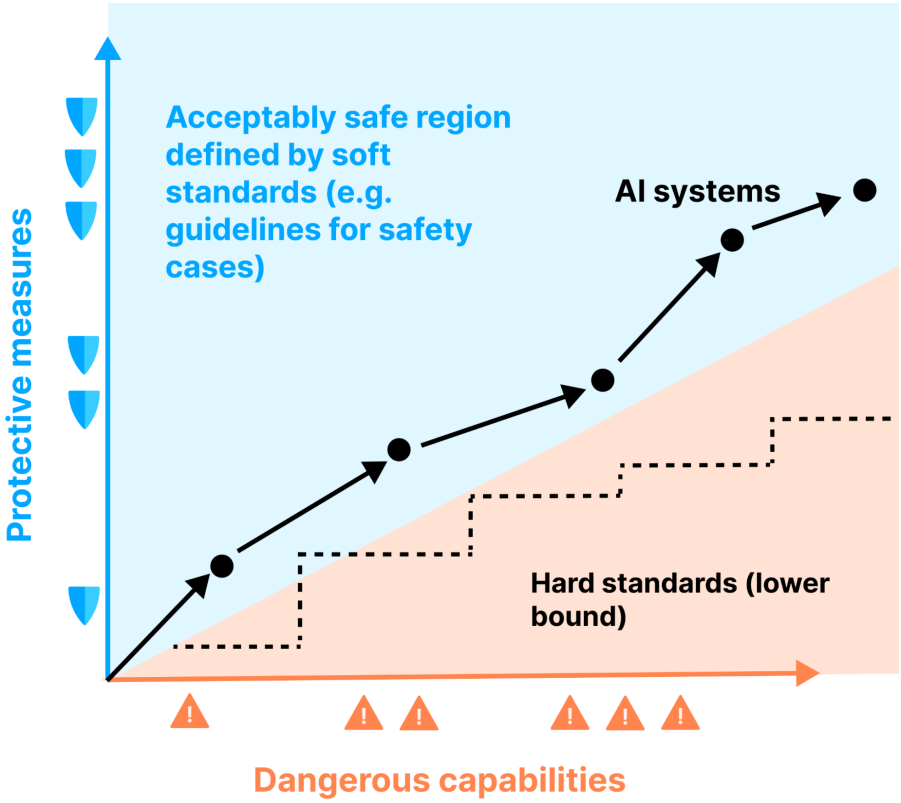


Figure 4: Hard standards are strict and objective, providing a 'lower bound' for safety while soft standards (e.g. guidelines for safety cases) can cover the rest of the distance by holistically setting regulatory expectations.

References

[1] U.S. Food and Drug Administration. Infusion pumps total product life cycle. Technical report, U.S. Food and Drug Administration, 2014.

[2] Radio Technical Commission for Aeronautics. Safety, performance and interoperability requirements document for the in-trail procedure in oceanic airspace (atsa-itp) application. Technical report, Radio Technical Commission for Aeronautics, 2008.

[3] Office for Nuclear Regulation. Safety assessment principles for nuclear facilities. Technical report, Office for Nuclear Regulation, 2020.

[4] Assurance Case Working Group. Goal structuring notation community standard, version 3. Technical report, Assurance Case Working Group, 2021.

[5] Charles Haddon-Cave QC. An independent review into the broader issues surrounding the loss of the raf nimrod mr2 aircraft xv230 in afghanistan in 2006, 2009.

[6] Duane Kritzinger. 5 - failure modes and effects analysis. In Duane Kritzinger, editor, *Aircraft System Safety*, pages 101–132. Woodhead Publishing, 2017.

[7] Nancy Leveson. White paper on the use of safety cases in certification and regulation. 2011.

[8] Mark Sujan, Ibrahim Habli, Tim P. Kelly, Simone Pozzi, and Christoper W. Johnson. Should healthcare providers do safety cases? lessons from a cross-industry review of safety case practices. *Safety Science*, 84:181–189, 2016.