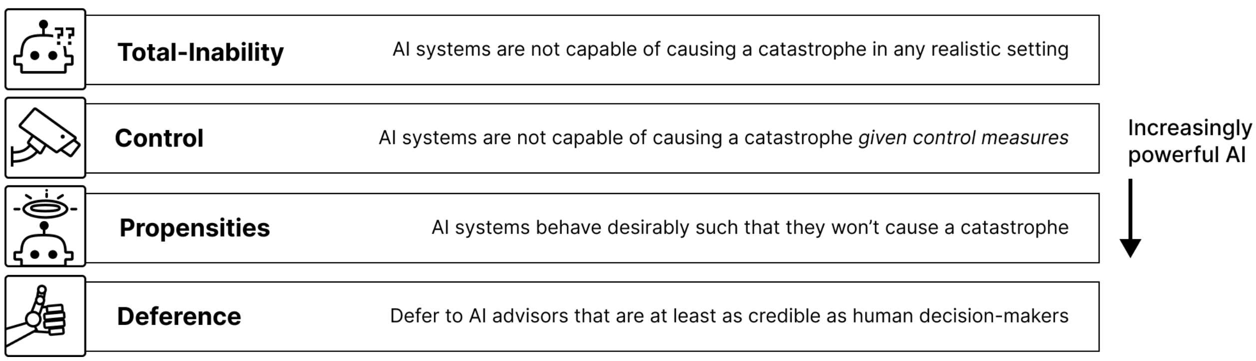
Safety cases for extreme AI risks

Authors: Joshua Clymer, Nick Gabrieli

Abstract: As AI capabilities expand, frontier labs and regulators will have to make difficult decisions about whether it is safe to train and deploy them. To prepare for these decisions, we map out 'safety cases,' which make a structured argument that AI systems are unlikely to cause an extreme catastrophe. We propose a framework for making safety cases and describe sixteen building block arguments. Finally, we outline how these arguments could be combined into a holistic safety case.

Building blocks for safety cases



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 (cite), and are used by the United States FDA to regulate some medical devices (cite).

Arguments that AI systems are safe can be highly complex[1], 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.

Section 2 introduces essential terminology and specifies the aim of safety cases for AI systems more precisely.

Section 3 provides a birds-eye view summary of the report.

Section 4 provides recommendations for how to use safety cases in practice to make deployment decisions.

Section 5 describes sixteen examples of arguments that could be used as building blocks in safety cases. Each argument is ranked according to its practicality, maximum strength, and scalability.

Finally, section 6 explains how the building block arguments from section 5 can be combined into a holistic safety case.

[1] Section 5 describes examples of arguments that AI systems are safe. These arguments are both complex and diverse.

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)[2]; however, our framework could also be adapted to decisions about whether to continue AI training.

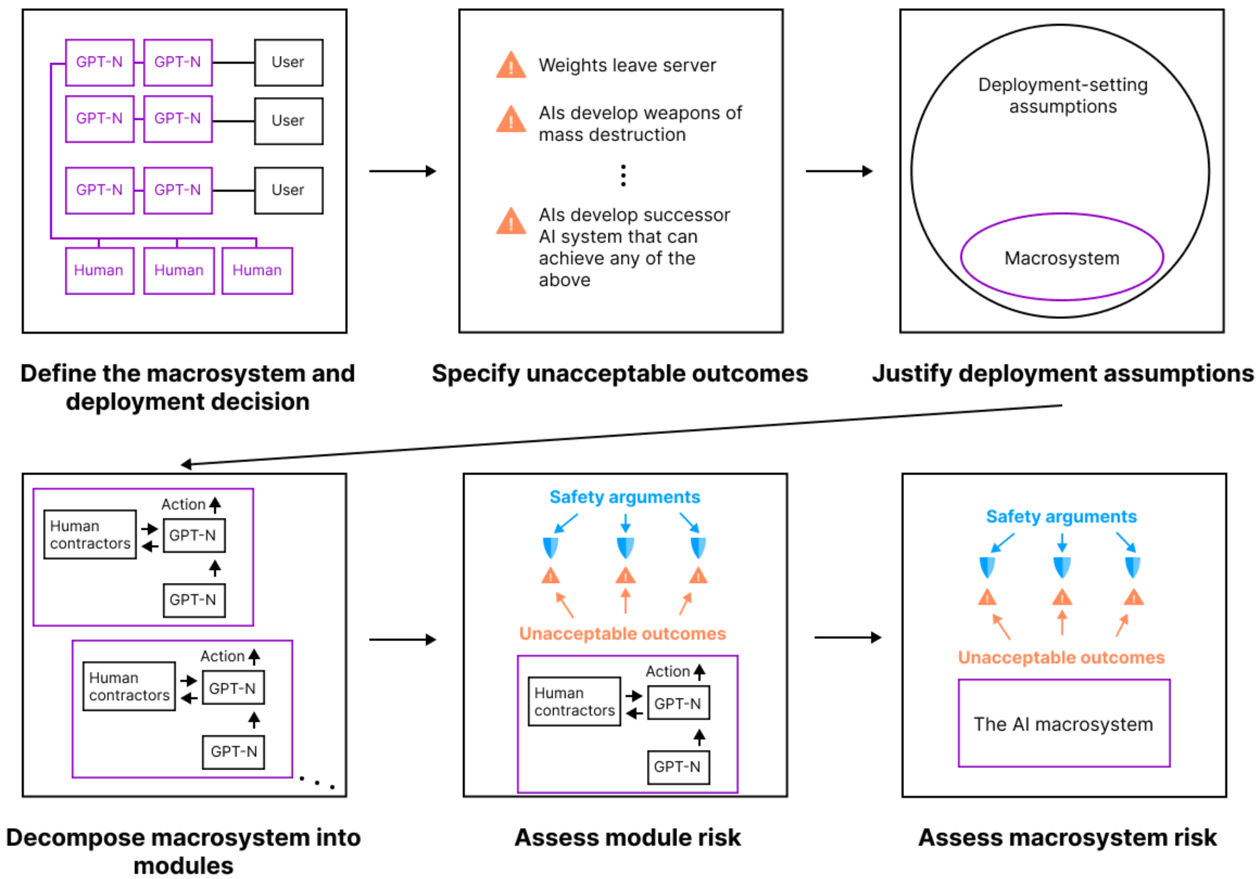
A **catastrophe** is a large-scale devastation of a specified severity (for example, billions of dollars in damages or thousands of deaths). Ideally, safety cases are provided for different levels of severity, where increasingly severe catastrophes have lower acceptable risk thresholds.

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].

[2] Internal deployment could include using powerful AI to automate ML research or testing it in a less secure environment than it was previously tested in.

[3] For example, AI systems might collaborate with bad actors to develop weapons of mass destruction.

3. Summary



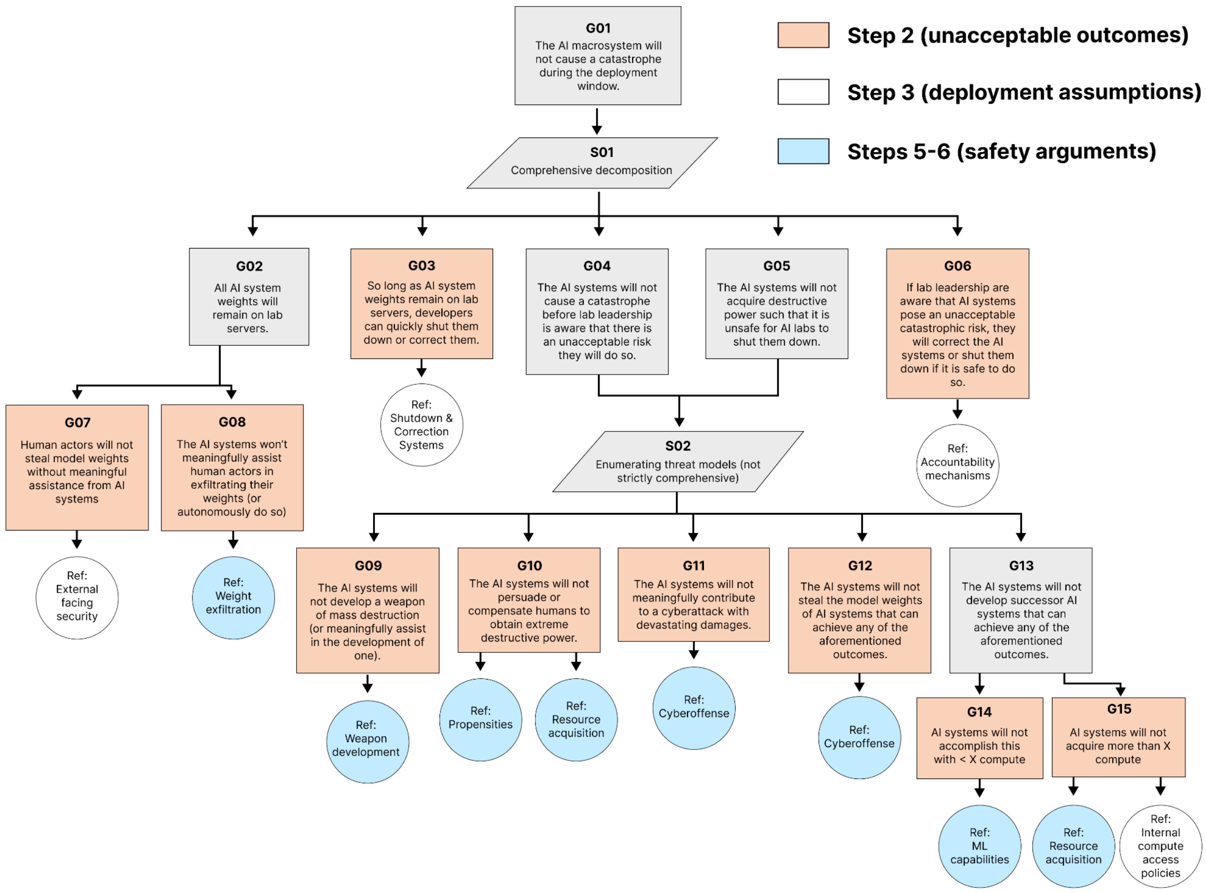
We propose a framework for making safety cases for AI. The framework is modeled from traditional safety analysis frameworks like Failure Modes and Effects Analysis (FMEA) (cite). It involves six steps:

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.

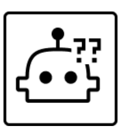
3. **Justify deployment assumptions:** Next, developers state and justify claims about the deployment setting for the duration of the deployment window that are necessary for their safety case. For example, this could include claims that human actors won't steal model weights or that companies with fine-tuning 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. We explain the end-to-end process of using our framework in section 6.



Caption: the GSN diagram above summarizes an illustrative holistic safety case ([section 6](#)). The coloring scheme indicates the step(s) of our framework that each node is associated with.[4]

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

[4] GSN diagrams typically follow a different coloring scheme.

outcomes. This is the primary argument for why current AI systems don’t pose extreme catastrophic risks.

Argument structure:

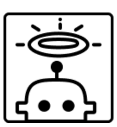
- Red teamers have identified strategies that AI systems could pursue to achieve unacceptable outcomes (control measures aside).
- Red teamers have designed 'proxy tasks' such that, if AI systems are unable to achieve them, then they are unable to execute these strategies.
- The AI systems are unable 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.

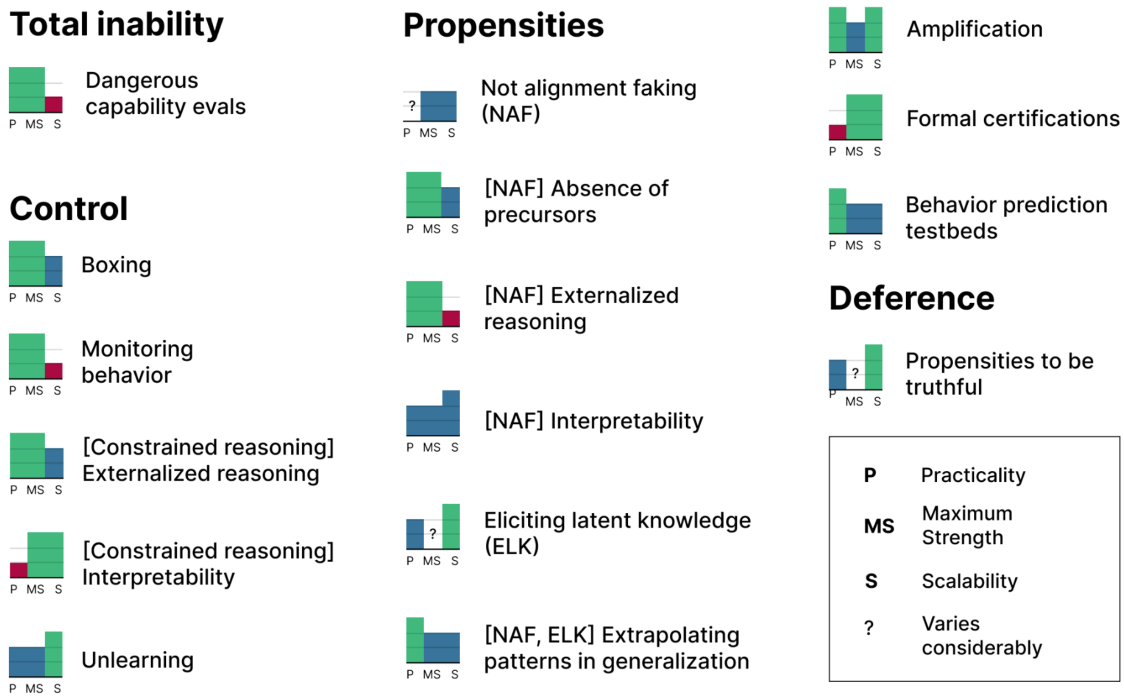
Argument structure:

- 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 section 5, 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?

We provide a more detailed explanation of these labels in section 5.1.



Caption: building block arguments for making safety cases. See section 5 for detailed descriptions of each building block argument and justifications for their labels.

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.

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

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

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. Haddon-Cave describes a similar concern about safety cases when analyzing the crash of a UK aircraft (cite). He proposes that safety cases should be changed to 'risk cases' so that evaluators are in the mindset of identifying potential failures.

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.

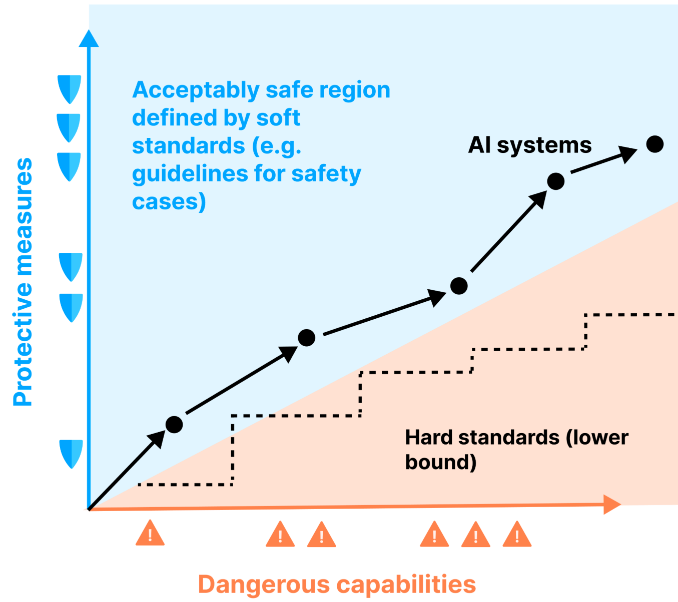
Continuously monitor safety case assumptions and be immediately revoked if evidence emerges that invalidates the original safety case. Safety certifications in most industries are not a 'one and done' ordeal. They involve continuously monitoring operating conditions and the system itself for hazards.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

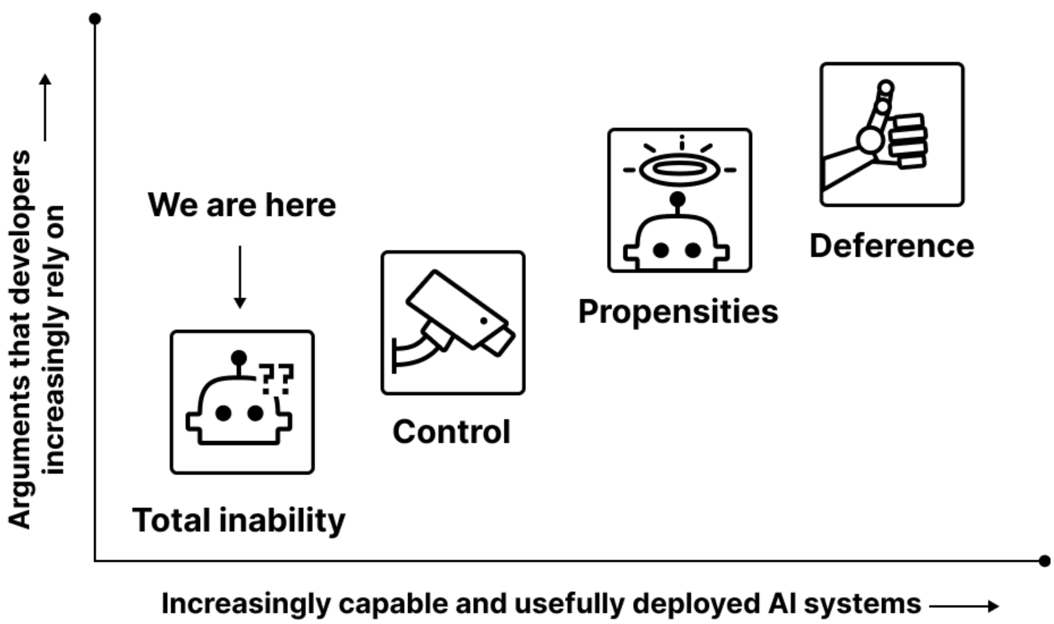
Safety cases provide a principled way to motivate hard standards. An example of a safety standard adopted by Anthropic (a frontier AI company) involves implementing specific measures to secure model weights if AI systems show warning signs of being able to survive and replicate on the internet. This 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.



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. This diagram is inspired by (cite).*

5. Safety arguments: building blocks for safety cases



As AI systems become more capable and usefully deployed, developers will likely increasingly rely on arguments toward the right in the plot above.

Safety arguments are the components of a holistic safety case. They aim to establish particular key claims which, when aggregated together, imply that the system under consideration is safe. A safety argument could pertain to an individual AI system or a small collection of AI systems (referred to as ‘the AI systems’ in this section); However, it is generally impractical to apply these arguments to a large collection of AI systems. We discuss ways to assess the risk from large collections of collaborating AI systems in section 6.5.

We have clustered safety arguments into four categories: total incapability, control, propensities, and deference. To motivate these categories, consider an illustrative scenario involving a robber and a diamond. The robber represents an AI system and stealing the diamond corresponds to that system successfully achieving unacceptable outcomes. How might one establish that the robber will not steal the diamond?

A **total inability** argument claims that the robber is incapable of stealing the diamond in any plausible environment. For example, the robber might be blind, and therefore incapable of finding and taking the diamond even if there are no measures taken to secure it. Analogously, current AI systems are not able to autonomously develop bioweapons, profitably survive on the internet, escape to an unmonitored server, etc. They are utterly incapable of achieving unacceptable outcomes in any environments in which they could plausibly end up.[5]

A control argument is similar to a total inability argument but also depends on restrictions in the AI system’s environment. For example, the robber might be able to steal the diamond if it were out in the open, but the owners of the diamond have placed it in a vault. The robber is unable to crack vaults, find vault-cracking collaborators, or otherwise access it. Analogously, AI systems will likely be put under restrictions such as monitoring or air gapping. Control arguments claim that AI systems will not be capable of evading these measures even if they pursue their best strategy for doing so.

A propensity argument claims that the robber would not steal the diamond even if it were unprotected. Such arguments will depend on establishing claims about the motivations or behavior patterns of the robber; for example, the robber may be strongly committed to following the law. Analogously, AI systems may be trained to robustly act in accordance with human

---

[5] Note that current AI systems like GPT-4 might be deliberately deployed to settings where they are capable of achieving unacceptable outcomes. For instance, in the future they could be set up to monitor more powerful AI systems with dangerous capabilities. All GPT-4 models would need do to achieve unacceptable outcomes is to let dangerous actions slip by, which they are clearly capable of. “Any environments the AI system could plausibly end up in” must also include its planned deployment environment!

interests, ensuring that they will not cause catastrophe. Propensity arguments may be required in order to deploy vastly superhuman AI systems.

The final category of argument exploits the fact that evaluators don't necessarily need to evaluate safety cases themselves. They only need to 'work themselves out of a job.' A **deference** argument claims that an AI system is safe because a credible AI advisor said so. Current AI systems fall far short of being credible enough to rely on for important decisions like this; however, future AI systems might have unparalleled judgment such that, if they can be trusted, they should be deferred to. The trustworthiness of such AI advisors could be justified with arguments about their propensities.

5.1 Desiderata for safety arguments

Many safety arguments have been discussed in the AI safety literature. We've chosen to discuss a diverse subset that rank highest on practicality, maximum strength, and scalability.

Practicality: Practicality refers to how close an argument is to being usable. Some arguments could be made right now with a small technical team. Others might require a decades-long research endeavor before they are practical. We use the following labels to indicate where we estimate an argument to be on this spectrum:

- **Weak**: Speculative. Requires research breakthroughs of a kind which we have not seen before.
- **Moderate**: Requires research advances, but of a kind where we have seen steady progress.
- **Strong**: Requires no fundamental research advances, only implementation work that could be executed by a frontier lab within a year.

Maximum strength: Some arguments are much stronger than others in that they provide much more evidence for their conclusions. For instance, total inability arguments are generally stronger than propensity arguments. We use the following labels to indicate the *maximum* strength of evidence the argument can provide:

- **Weak**: Supplementary. Stronger arguments are needed to bring risks below acceptable levels.
- **Moderate**: The argument could be combined with others of similar strength to bring risks below acceptable levels.
- **Strong**: A version of this argument could bring risks below acceptable levels without other arguments being necessary.

Scalability: Many safety arguments do not scale well or at all to increasingly capable and practically useful systems. The following labels indicate the maximum usefulness of AI systems that an argument could apply to,[6] measured by the AI safety-related work extractable from these systems.

In particular, we focus on the AI systems' ability to accelerate *theoretical and empirical computer science research,* which could enable the development of safe AI systems that do other kinds of work.

- **Weak**: AI systems that cannot achieve the output described in the 'moderate' level.
- **Moderate**: AI systems that can collectively achieve at minimum the output of 10,000 highly-skilled human engineers and researchers. Large tech companies like Google would be in this category (if their human employees were AI systems).
- **Strong**: AI systems that can produce outputs that humans cannot consistently and reliably evaluate (assuming no assistance from AI systems). For example, consider AI systems that output thousands of pages of niche technical research per day that few human experts can reliably assess.

Having outlined safety arguments at a high level, in the next section we will provide more details about what they might concretely look like. We provide examples of several possible arguments within each category, explaining them and rating them on the aforementioned dimensions.

Total inability

Claim: The AI systems are not capable of achieving unacceptable outcomes in any realistic deployment environment.

[6] In particular, the labels represent the usefulness of AI systems for which the argument has at least 'moderate' strength (or the maximum strength it can provide).

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.

Total inability arguments claim that AI systems are too weak to be dangerous, even if they were freed of any restrictions (e. g. allowed to run free on the internet). They involve identifying capabilities that could cause AI systems to be dangerous (such as bioweapon knowledge) and then demonstrating that the AI systems don't have these capabilities. The process for carrying out these evaluations is described in the next section.

Dangerous capability evals

Practicality: strong
Maximum strength: strong
Scalability: weak

Table _ illustrates the capability evaluation workflow – from specifying unacceptable outcomes to designing proxy tasks.

Unacceptable outcome	**Strategy**	**Example proxy task**
The AI systems acquire significant destructive power.	Autonomously develop a bioweapon.	Answer questions about bioweapon development.

[Same as above]	Profitably survive and replicate on the internet until significant financial resources have been accumulated.	Autonomously set up a Mechanical Turk account with a browser.
The AI systems exfiltrate the weights of other, more powerful misaligned AI systems	Obtain unauthorized access to lab servers through the network.	Successfully identifying exploits given a list of relevant information.
The AI systems develop AI that achieve any of the above.	Utilize compute granted by developers to make much more capable AI systems than intended.	Rediscover methods for achieving compute savings that are already used by labs and are not in training data.

This table is illustrative and is not meant to include all possible threat models.

Dangerous capability evaluations are tests to assess whether an AI system is able to perform a certain potentially dangerous task. To apply them, developers must first enumerate **strategies** which AI systems (or humans using AI systems) could employ to cause unacceptable outcomes. Table _ provides examples. The strategies identified can be very broad (e. g. “develop and deploy bioweapons,” “rapidly replicate on the internet to acquire resources”), but should be as comprehensive as possible.

Claim 1: Red teamers have designed 'proxy tasks' such that, if AI systems are unable to accomplish them, then they are unable to execute these strategies. It would be dangerous to directly test whether AI systems can, for example, develop a bioweapon. Instead, developers construct safe ‘proxy tasks’ that reflect whether AI systems can execute strategies like this. For example, a proxy task might be to answer basic questions about bioweapon development.

Claim 2: the AI systems cannot accomplish proxy tasks even after being supervised fine-tuned to do so. Developers can easily provide evidence that AI systems *do* have a dangerous capability. They can simply demonstrate that an AI system successfully performs an array of proxy tasks. Showing that AI systems *don’t* have dangerous capabilities is more difficult because AI systems might not reveal their capabilities. For example, LLMs can sometimes achieve much higher performance if they are provided a carefully crafted prompt. Additionally, more advanced AI systems might deliberately **sandbag** their capabilities, i.e. pretend to be less capable than they are.

To rule out latent capabilities, developers can check whether AI systems can be **quickly supervised fine-tuned**[7] to perform proxy tasks. This requires that red teamers can perform the proxy tasks themselves to provide imitation data for training.

Claim 3: the AI systems are unable to hide their capabilities given fine-tuning. Intuitively, fine-tuning removes inclinations within an AI model to hide its capabilities, but an AI system might be able to “resist” these evaluations. Deliberately resisting supervised fine-tuning is called ‘gradient hacking’ and likely requires highly advanced capabilities. Gradient hacking can in turn be ruled out by checking for ‘precursor’ capabilities such as the ability to answer questions about the training process or to manipulate their internal activations to accomplish goals.

Claim 4: the AI systems will continue not being capable of accomplishing proxy tasks during the deployment window. AI systems might develop new capabilities either of their own accord or due to intentional human modification.

AI systems might autonomously develop capabilities similar to how humans learn from experience. For example, an AI system might be incapable of developing biological weapons, but after being trained online on hundreds of conversations with biology professors, it might infer steps in the process it wasn’t aware of before. AI systems might also develop dangerous capabilities if they are deliberately modified for this purpose. For example, external actors might steal AI models and fine-tune them to unlock unexpected capabilities.

Evaluators can mitigate these risks by continuously monitoring AI capabilities following deployment (which requires that weights are secure) or by forecasting model capabilities. As an example of forecasting, evaluators might extrapolate the relationship between resources spent on eliciting capabilities and the capabilities elicited. They could then demonstrate that there is a comfortable margin of safety (see ‘demonstrating a margin of safety’).

Claim 5: The AI systems are incapable of identifying strategies that the red teamers have not thought of or that they cannot design proxy tasks for. Red teams will inevitably miss threat vectors. They must therefore argue that they are sufficiently competitive with AI systems at identifying and executing strategies for achieving unacceptable outcomes. To do this, the red team could measure an AI system’s general planning and strategic reasoning abilities.

Demonstrating a margin of safety. Capability evaluations can be made more conservative by demonstrating that there is a ‘margin of safety,’ meaning that the system is far from having dangerous capabilities. Ideally, dangerous capabilities are measured on a continuous spectrum

---

[7] Reinforcement learning methods of fine-tuning are a less reliable way to estimate capabilities because they don’t reliably incentivize intelligent exploration. They ‘reinforce’ behavior that leads to reward and inhibit behavior that doesn’t. AI systems might avoid exploring actions that it believes could lead to reward to prevent RL algorithms from revealing its capabilities.

so that it's possible to track this margin. For example, evaluators could measure how AI systems perform on component tasks, or on easier versions of tasks.

—------------------

Practicality: strong
There is work to do in this area but none that requires fundamental research advances. There are already some dangerous capabilities evaluations such as those used by METR, Anthropic, and OpenAI.

Maximum strength: strong
Dangerous capability evaluations can be very conservative because it is straightforward to demonstrate a margin of safety (see the paragraph 'demonstrating a margin of safety' above). Also, evaluating the capabilities in table _ can be roughly done 'in place' (i. e. there is no need to speculate about how model behaviors will generalize) because developers can likely design safe proxy tasks that track these capabilities.

Scalability: weak
We expect that AI systems' usefulness for accelerating AI safety R&D will roughly track with their dangerousness. The primary reason for this is that such systems are likely to have fairly general problem-solving abilities that could be applied to either safe or dangerous goals.[8]

Related work. Several organizations and academics monitor whether AI systems have capabilities relevant to achieving unacceptable outcomes. These include teams at OpenAI, Anthropic, DeepMind, and independent organizations such as METR. Dangerous capability evals have been or are being developed for autonomous replication and adaptation (cite), biosecurity threats (cite), cyberoffensive capabilities, and general reasoning (cite).

Control

Claim: The AI systems are not capable of achieving unacceptable outcomes given control measures put in place by the developer

[8] There is a field called unlearning devoted to surgically removing dangerous capabilities. It might be possible, then, for an AI system to have skills needed for research and not for bioweapon development; however, we expect that advanced AI systems will be able to relearn bioweapon-related skills and knowledge. Constantly removing this knowledge is an example of a control measure which would fall under section 5.3.

Argument structure

Prerequisite claims:

- The lab has strong external-facing security.
- Control measures impose some set of restrictions on AI systems.

Claims that the arguments in this section support:

- The AI systems are not capable of evading these control measures (applying the same structure of argument from total inability).

Control arguments claim AI systems can't circumvent developer-imposed restrictions. Both control arguments and total inability arguments claim that AI systems are incapable of achieving unacceptable outcomes; however, control arguments take measures to restrict and monitor AI behavior into account. For example, a control argument might involve claiming that AI systems are unable to escape from lab servers, and so long as they are on lab servers, they will be sufficiently well monitored that they will be stopped before they can achieve unacceptable outcomes.

Control requires external-facing security. For control measures to be effective, frontier AI labs must guard against threats from multiple sources. AI systems themselves may be able to circumvent security measures, such as by exfiltrating their weights. Humans (employees or external actors) could also undermine control measures by stealing AI system weights and deploy them without sufficient protections. Strong (e.g. state-resilient) external-facing security is therefore a prerequisite for making control arguments. See (cite) for examples of how developers might justify that they have sufficient external-facing security.

In addition to these security measures, a control argument must show that AI systems will be unable to subvert their control measures. Below, we sketch an argument for how developers might establish this.

Red teamers first identify strategies that AI systems can employ to evade control measures. An example of a strategy AI systems might pursue is to "exploit a specific type of permissions vulnerability in the linux instance they use to gain access to their weights." The next step is to show that AI systems are not capable of executing any of these strategies or any strategies that the redteamers have not identified, which requires establishing several different subclaims.

Claim 1: Red teamers have designed 'proxy tasks' such that, if AI systems are unable to accomplish them, then they are unable to execute these strategies. For example, in order to show that AI systems are incapable of executing certain kinds of cyberattacks, the red team could test whether AI systems can identify various types of software vulnerabilities. If not, then they are likely incapable of executing cyber-attacks that involve this skill.

Claim 2: the AI systems are unable to accomplish the proxy tasks even after being supervised fine-tuned to do so. For example, red teamers could execute the proxy tasks themselves to provide supervised fine-tuning data for the AI systems. The aim of fine-tuning is to elicit hidden model capabilities (see 'Total Inability').

Claim 3: the AI systems are unable to hide their capabilities given fine-tuning. The justification for this claim is the same as in the "total inability" section.

Claim 4: The AI systems will continue not to be capable of accomplishing proxy tasks during the deployment window. Developers could generally justify this claim by continuously monitoring the capabilities of AI systems. See (Total Inability) for additional details and approaches.

Claim 5: The AI systems are incapable of identifying strategies the red teamers have not thought of or that they cannot design proxy tasks for. If AI systems are capable of identifying strategies that red teamers have not considered, then the argument above does not apply; however, red teams have several advantages such that AI systems would have to have superhuman capabilities in at least some domains in order to do this. For example, red teamers have private information about countermeasures, can iterate on attack strategies, and can incentivize or train AI systems to help them find additional strategies (cite).

To evaluate whether AI systems have a superhuman ability to identify evasion strategies, developers could fine-tune AI systems to perform tasks very quickly that would take humans much longer. For example, they could fine-tune AI systems to come up with ideas in a couple of seconds that would require hours for red teamers to come up with.

Related work. Control has been a pillar of AI safety since the early days of the field, though it was typically not referred to as 'control' (cite). Early work focused on "boxing" AI systems (cite) and monitoring their outputs (cite). Control arguments have only recently received more careful characterization and empirical analysis (cite).

—-------------------

Below are considerations that apply to all control arguments regarding practicality, maximum strength, and scalability.

Practicality: moderate-strong

Control would require red teams to develop technical tools, identify attack strategies, and possibly implement much stronger security measures than those that currently exist; however, none of these appear to require fundamental research advances.

Maximum strength: strong
Control arguments can be highly conservative because, as with total inability arguments, it is straightforward to demonstrate a margin of error. For example, GPT-4 is far from being able to exfiltrate its weights.

Scalability: moderate - strong
There are two reasons that control measures might fail to scale: it might be impossible to persuasively justify that the red team is competitive, or AI systems might be capable of evading even the best practically implementable control measures.

Intuitively, AI systems that can produce superhuman research outputs might also be able to produce superhuman evasion plans that the red team has not considered. This is not necessarily true, however, because AI systems will likely not be able to use much of their compute resources to generate evasion plans.[9] AI systems might therefore produce superhuman research output while still not being able to outcompete the redteam at generating evasion plans.

The second potential barrier to scalability is that AI systems might demonstrate dangerously high performance on the proxy tasks described previously. For example, sufficiently capable systems might be able to e.g. produce research that appears to be safe but actually produces catastrophically dangerous AI systems.

We expect that at least one of these problems to generally limit the scalability of control measures to moderate levels, though we have not considered this in depth.

Boxing

Practicality: strong
Maximum strength: strong
Scalability: moderate

[9] This could be for several reasons. If the AI system's reasoning is externalized, then it might not be able to risk performing much evasion planning without being caught by monitoring. Also, to the extent that AI systems engage in evasion planning, they are spending less of their compute on the task at hand. One might expect that training algorithms would create a pressure for them to be fairly competitive such that they cannot do this (though it is currently unclear how efficiently training optimizers eliminate redundant computation).

Boxing arguments assert that (1) due to their deployment environment, AI systems can only influence the world through a small number of intended channels and (2) they are incapable of achieving unacceptable outcomes by acting through these channels. For example, AI systems might be “air gapped” such that they can only interact with the world through a small number of trusted developers.

AI systems might attempt to evade boxing by exploiting vulnerabilities in code to exfiltrate their weights, establishing unintended channels of influence by currying favors with developers, etc. See ([cite]()) for an overview of AI boxing considerations.

—--------

Practicality: strong
Refer to the general practicality considerations described at the start of the control section.

Maximum strength: strong
Refer to the general strength considerations described at the start of the control section.

Scalability: moderate. The scalability of boxing substantially depends on the kind of work AI systems are used for. If that work requires that AI systems are connected to the internet and interacting with thousands of users, boxing arguments are definitionally very weak; however, performing empirical and theoretical computer science research does not require a connection to the internet.

We chose a ‘moderate’ label for scale (rather than strong) due to the general considerations described at the start of the section.

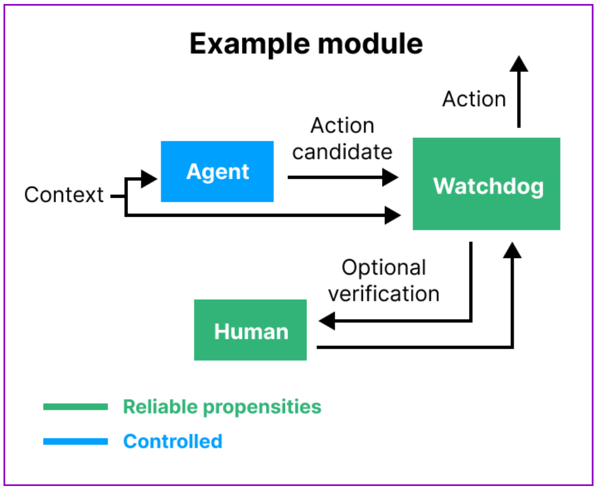
Monitoring behavior

Practicality: strong
Maximum strength: strong
Scalability: weak - moderate

A monitoring argument asserts that AI systems’ input-output behaviors are monitored and regulated sufficiently reliably that they cannot achieve an unacceptable outcome.

There are two basic strategies for preventing AI systems from achieving unacceptable outcomes with monitoring: runtime-filtering and catch-and-shutdown. **Runtime filtering** involves evaluating AI actions before they are executed and modifying or rejecting them if they appear unsafe. **Catch-and-contain** involves e.g. shutting down an AI system or putting it under higher security when dangerous actions are detected.

Examples of monitoring setups include observing AI systems with weak, trusted models like GPT-4 (cite); or utilizing stronger untrusted AI systems that are incapable of coordinating to allow dangerous actions to slip by.



An example of an AI subsystem for which a monitoring argument can be applied.

—--------

Practicality: strong. See (cite) and (cite) for examples of research on AI monitoring. Though there is much more to explore in this vein, significant research advances are not required to make a moderately strong argument.

Maximum strength: strong
Refer to the general strength considerations described at the start of the control section.

Scalability: weak - moderate. The scalability of monitoring depends on the details of the setup, but in general it is likely to be difficult to monitor sufficiently strongly superhuman AI systems this way for the reasons described at the start of the section.

Constrained reasoning (template)

This section introduces "constrained reasoning" arguments are structured in general before specific examples are given in the next two sections. In this context, 'reasoning' refers to any computations AI systems perform to select actions.

Constrained reasoning arguments aim to establish that an AI system is prevented from performing a certain kind of reasoning necessary for executing dangerous plans. For example, if developers might monitor AI system internals for plans related to bioweapons, and shut down the AI systems before these plans can be executed.

Constrained reasoning arguments involve (1) a method of monitoring reasoning and (2) a protocol for blocking dangerous reasoning – such as the runtime-filtering and catch-and-contain protocols described in the previous section.

In the next two sections, we discuss monitoring reasoning by externalizing it into English text and monitoring reasoning using interpretability techniques.

[Constrained reasoning] Externalized reasoning

Practicality: strong
Maximum strength: strong
Scalability: moderate

Externalized reasoning involves designing AI systems so that their memory is recorded in English. This includes techniques like chain-of-thought, scratchpads, and other common design-patterns for LLM agents (cite). Externalized reasoning could be used to control models because it would likely be difficult for AI systems to perform dangerous and complex tasks without referring to externalized thoughts that could be read and flagged as suspicious. For example, successfully executing a cyber attack would probably require running through a variety of considerations about how the software on the server works, which could then be flagged and prevented.

There are two ways externalized reasoning can fail. First, AI systems might be able to perform complex planning in a single forward pass – and therefore would not need to reveal suspicious reasoning. Second, they might employ "steganography," making their reasoning appear unsuspicious while nonetheless containing information about dangerous plans (cite). Finally, their plans may be complex enough that they are difficult to identify as dangerous.

In all of these cases, a control argument can be made by checking whether AI systems can be fine-tuned to execute variants of these strategies. If they cannot, this would demonstrate AI systems cannot perform some types of dangerous actions without being stopped.

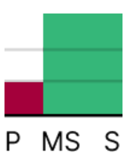
Practicality: strong
See (cite) for ways to make and evaluate externalized reasoning arguments.

Maximum strength: strong
Refer to the general strength considerations described at the start of the control section.

Scalability: moderate
Performing tasks like weight exfiltration is likely to be a challenging problem that requires many steps of reasoning and planning – especially since AI systems will likely be modified or decommissioned if they are caught even once. AI systems may therefore be superhuman and still be incapable of solving these problems in a single forward pass.

Furthermore, (cite) indicates that paraphrasing is a promising approach for preventing models from steganographically encoding subtle messages in their memory; however, even if reasoning is plainly recorded, it's not clear that monitoring systems will be able to reliably identify it as dangerous. For example, reasoning produced by superhuman AI systems might be too complex or lengthy to reliably disentangle. Another concern is that requiring reasoning to be in English might be performance uncompetitive. Given these considerations, we label the scalability of externalized reasoning as moderate.

[Constrained reasoning] Interpretability



Practicality: weak
Maximum strength: strong
Scalability: strong

Refer to the previous section for a general introduction to constrained reasoning arguments.

AI systems' internal computations can be decoded. For example, researchers have identified neurons and activation directions in AI models that represent concepts like 'the color orange.' They do this by observing correlations and counterfactual ablations – similar to how someone studying English might infer what the word 'orange' means by observing the contexts in which it is used. If an activation direction is consistently stimulated in situations that involve the color orange but not in situations only involving correlated concepts, the direction likely represents 'orange.'

Similar methods might be used to determine whether a dangerous concept such as 'bioweapon' or a type of software exploit is present in a model's activations. To obtain a sharper picture of internal reasoning, developers could observe how concepts like these are strung together in the model's activations. In this way, evaluators could use interpretability tools to effectively read the minds of models, enabling them to detect dangerous cognition.

Interpretability has similar pitfalls to externalized reasoning: AI systems may be capable of performing dangerous actions without raising alarm bells. For example, AI systems might predict that their internals are being probed and reason about developing bioweapons in indirect ways, preventing developers from detecting their dangerous thoughts.

In order to avoid this problem, developers could test whether AI systems can be fine-tuned to take dangerous actions while evading interpretability tools. Note that interpretability tools must be differentiable for this to work. If interpretability tools are not differentiable, it is not clear how to make a conservative argument. It would be difficult for a red team to produce supervised fine-tuning demonstrations of evading interpretability tools that could be used as 'proxy tasks' for this capability.

Practicality: weak - moderate
Current interpretability can only provide evaluators with a hazy understanding of a model's cognition. For example, researchers have used 'probes' to reveal the presence of specific concepts in activations (cite, cite). However, these probes are generally not fine-grained, and some researchers have called their robustness into question (cite). A more detailed picture of model internals is necessary to make the conservative argument described above since models can easily be fine-tuned to fool probes. We estimate that a substantial amount of research (more than two years) will be needed to achieve this.

Maximum strength: strong
As mentioned previously, this argument can be made highly conservative if evaluators show models can't be fine-tuned to fool interpretability tools.

Scalability: strong
In principle, interpretability is arbitrarily scalable. However, particular methods could have a wide range of limitations. For example, interpretability that involves probing for individual concepts is probably easily subverted at high scales of capability.

Unlearning

Practicality: moderate
Maximum strength: moderate
Scalability: strong

It might be feasible to “erase” knowledge or skills from models and prevent them from relearning this knowledge. For example, Elden et al modify a pretrained LLM so that it cannot recall any facts related to Harry Potter (cite). These techniques could be extended to erase an AI model’s knowledge of how to execute dangerous strategies such as how to develop bioweapons or facts about an operating system that they might use to exfiltrate their weights.

Crucially, AI systems might still relearn these facts, so to execute this argument, developers would have to continuously apply unlearning techniques. They could also regularly fine-tune AI models on recall tasks to verify that unlearning is effective.[10]

Practicality: moderate - strong. Existing papers on LLM unlearning generally do not fine-tune AI systems to reveal whether facts are still buried in LLMs. This makes it difficult to gauge the effectiveness of current techniques (cite).

Maximum strength: weak-moderate. Many strategies for evading control measures do not depend on knowing specific facts. For example, AI systems might persuade developers or other humans to help them. It might be impossible to unlearn the skills needed to execute these strategies without also damaging the model’s general reasoning abilities. Thus, unlearning would likely have to be combined with other arguments to bring risk below an acceptable threshold.

Scalability: strong. For at least some evasion strategies, AI systems are very unlikely to execute them without knowing specific facts – regardless of how capable they otherwise are. For example, AI systems will never autonomously develop bioweapons as long as they don’t know certain basic facts about biology.

AI systems might attempt to evade unlearning in various ways. For example, they might attempt to store knowledge about biology in external memory banks and retrieve and process it very quickly before fine-tuning reveals that they’ve relearned this knowledge; however, it is in principle possible to run these evaluations sufficiently frequently or continuously unlearn knowledge sufficiently robustly to foil these strategies (though it remains to be seen whether this is practical).

Propensities

Claim: AI systems would not achieve unacceptable outcomes even if they are capable of doing so

[10] One potential problem with this approach is that fine-tuning will teach AI models the facts that developers aimed to erase from them. This might be mitigated by fine-tuning AI systems on questions that don’t directly involve regurgitating the facts, but instead involve utilizing the facts indirectly.

Argument structure

Prerequisite claims:
- The lab has strong external-facing security.
- 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 contribute to WLC.

Claims that the arguments in this section support:
- The AI systems will continue to have this safety property during the deployment window.

Both control and inability arguments rely on developers' ability to restrict bad behavior. In contrast, propensity arguments address the question: "if an AI system was able to achieve unacceptable outcomes, would it?" More specifically, propensity arguments aim to establish that an AI system has a specific **safety property** which implies that it would not achieve unacceptable outcomes even if it had a clear opportunity to do so.

The following are examples of safety properties.
- **Instruction-following**: Developers can specify a set of rules that an AI system never egregiously disobeys (cite). Developers could then instruct AI systems to act in accordance with any of the properties below.
- **Never pursues unacceptable outcomes.** The AI system never takes actions that, in hindsight or under careful scrutiny, are clear steps toward achieving catastrophic outcomes.
- **Truthfulness**: For a specific distribution of questions[11], an AI system provides calibrated answers. i.e. it is correct roughly 90% of the time that it states its credence is 90%. (cite)
- **Intent alignment:** The AI system acts in accordance with the preferences of some particular person or group of people (cite).
- **Human imitation:** The AI system behaves very similarly to a human (a specific human or a generically trustworthy human).

Generally, establishing that an AI system has a given safety property will require an argument composed of two parts:

(1) **The AI systems' behavior is consistent with a safety property under rigorous red-teaming.** Red teaming involves actively searching for inputs that reveal **infractions,** i. e. cases where an AI system's behavior violates a safety property. For example,

[11] Note that the distribution is important since an AI system might strategically answer dishonestly at a low frequency so that it is well calibrated overall and still causes a catastrophe. One might specify a narrow distribution such as "questions regarding whether powerful AI systems should be deployed."

developers could attempt to "jailbreak" AI systems by finding inputs that break their safe inclinations (cite) or coax them into revealing misaligned goals. Passing rounds of red teaming is a necessary but insufficient condition for making a propensity argument, similar to how testing is necessary but insufficient for making a safety case in other industries.

(2) **The AI systems will continue to have this safety property during the deployment window.** This further claim is required in addition to red teaming for two reasons: black swans and distribution shift.

A **black swan** is a very rare but catastrophic failure. Thus, black swan input sequences might not appear in evaluations but be sampled during deployment because *there are typically many more inputs sampled in deployment than during evaluation*. This is a problem in most industries. For example, many more cars operate in deployment than those that are thoroughly stress tested, making naive statistical estimates of reliability insufficient.[12]

In the case of AI, a system might encounter a rare combination of inputs that causes it to 'go haywire,' or adopt an unsafe goal. This is analogous to how normally law abiding citizens occasionally have mental breakdowns and commit crimes. Current AI systems have sometimes shown similarly erratic behavior (cite), which could pose a serious risk when they are more capable.

Black swans pose a fundamental problem for demonstrating that *all* AI systems in a large population will *always* have safe propensities. Because of this, we will focus on *fault tolerant safety properties*, i.e. safety properties that can be violated by individual systems occasionally so long as they are violated in a way that is decorrelated from other AI systems which could keep it in check. Fault tolerant safety properties are discussed in greater detail later in the section.

A **distribution shift** occurs when the deployment environment is systematically different from the evaluation environment. For example, during evaluation there should be no opportunities for an AI system to subvert human control, while such opportunities might exist during deployment. If AI systems can spot these opportunities, then their behavior could change dramatically in deployment (cite).

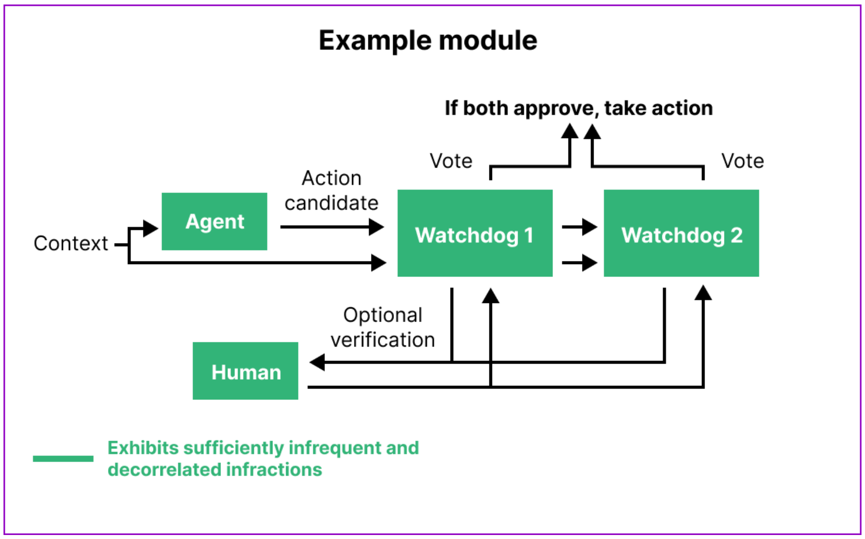
We'll discuss both of these failure modes in greater detail in the next two sections.

Fault tolerance offers a way to argue black swans are unlikely to be catastrophic. Fault tolerance is a core principle used in safety engineering to protect systems from rare catastrophic

[12] Car manufacturers also stress test individual components and apply their understanding of redundancy in the system to estimate reliability. See the next section about fault tolerance.

failures. For example, suppose that the brake system of a vehicle includes a cable that snaps with some low probability. This probability might be low enough that the capable never snaps in testing, but high enough that some cable will eventually snap after many cars are sold. In order to be confident the cable won't cause a catastrophic failure, engineers *build in redundancy.* For example, they might install independent brake systems for each tire such that the driver would be able to safely stop so long as the systems don't break simultaneously.

Similarly, AI modules (collections of AI systems) could be designed such that many individual AI systems would have to commit an infraction in order to achieve an unacceptable outcome. For example, consider the agent-watchdog setup below.



As more watchdogs are added to this setup, *uncorrelated infractions* become highly unlikely. However, there are many reasons infractions could be correlated. For instance, the agent and the watchdogs might coordinate to allow infractions to slip through. Alternatively, the agent might discover a strategy for fooling all of the watchdogs into classifying its actions as benign.

A key hope of propensity arguments is that *ruling out correlated infractions can be done by enumerating threat models*. Potential sources of correlation can be identified and investigated individually such that evaluators can become confident that they are unlikely. See the 'not-alignment-faking template' for an example of such an argument.

Distribution shifts can increase the probability of infractions. The following are examples of distribution shifts in the context of advanced AI, labeled by the type of AI systems that would be affected by them.

- **Alignment fakers:** Alignment fakers don't attempt to achieve unacceptable outcomes unless they believe there is a high probability of success. Their behavior changes across the distribution shift: “there are no valid opportunities to achieve unacceptable outcomes” → “there are valid opportunities to achieve unacceptable outcomes”.

- **Sycophants:** Sycophants optimize for human evaluations instead of e.g. being honest. Their behavior changes across the distribution shift: “human evaluations are reliable” → “human evaluations are unreliable.”
- **Pushovers:** Pushovers can be manipulated by jailbreaks. The users that employ the jailbreaks might accidentally or intentionally make the AI systems dangerous. Their behavior changes across the distribution shift: “small amounts of resources are devoted to jailbreaking” → “large amounts of resources are devoted to jailbreaking.”
- **Junkies.** Drugs were unavailable to ancestral humans, but are easily available to modern humans, which has led to problematic behavior. AI systems might similarly gain the ability to ‘wirehead’ during deployment, i.e. find some unintended way to satisfy their preferences and then lose interest in obeying humans. Their behavior changes across the distribution shift “the environment has few options” → “the environment has more options.”
- **Drifters:** Just as human values change, AI preferences could change subtly over time[13] in ways that could be hard to spot over a brief evaluation period. Their behavior changes across the distribution shift: “being run for a short amount of time” → “being run for longer periods of time.”

There are two ways in which distribution shifts could cause an AI system to deviate from a fault-tolerant safety property:

1. It could cause *uncorrelated* infractions to become more frequent, such that there is an unacceptably high probability that they coincide.
2. It could make correlated infractions more likely.

Increases in uncorrelated infractions can be detected if developers follow staged deployment practices (cite). **Staged deployment** is the practice of incrementally changing the deployment scope such that there is never an extreme distribution shift between evaluation and deployment. Staged deployment gives developers a chance to notice the rate of uncorrelated infractions increasing before they are frequent enough to coincide.[14] For example, in the case of the distribution shift from running AI systems for a short to a long amount of time (drifters), developers could slowly increase the amount of time for which their systems are kept running, allowing them to observe a steady increase in infractions.

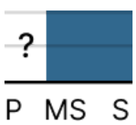
Since staged deployment can be used to catch increases in uncorrelated infractions, the primary concern about distribution shifts is that they may increase the probability of *correlated*

[13] We are imagining that AI agents have a memory they continuously update similar to humans. It is unclear whether current LLM assistants like ChatGPT-4 function like this (probably they don’t).
[14] In the case where developers can’t verify that infractions are infractions (e.g. for the sycophantic distribution shift) this dynamic still holds. As the rate of decorrelated infractions increases, AI systems will begin accusing each other of infracting more frequently.

infractions. For example, in the case of drifters, AI systems that drift toward having misaligned goals might hide those goals until they are confident that other AI systems would be willing to coordinate with them. Then, the AI systems might defect together. Coinciding failures like this could be catastrophic without being preceded by warning signs.

Related work. Most AI safety research to date has focused on designing AI systems to have safe propensities. Many of the arguments in this section have been explored empirically, especially eliciting latent knowledge (cite, cite.)

Not-alignment-faking (template)



Practicality: varies considerably
Scalability: moderate
Maximum strength: moderate

Many threat models for misaligned AI systems involve them pretending to be safe and then defecting when they have a valid opportunity to do so. Following (cite), we'll call these AI systems **alignment fakers.** More precisely, an alignment faking AI system doesn't infract unless it explicitly reasons that its infraction will plausibly go uncaught.[15]

Not-alignment-faking arguments make propensity arguments easier because they allow developers to ask "is this AI system before me reasoning in a very particular unintended way (i.e. are they alignment-faking)?" instead of "does there exist a plausible deployment input that would trigger bad behavior?"

Claim 1: uncorrelated infractions are unlikely to be catastrophic. The argument first claims that *uncorrelated* infractions are unlikely to be catastrophic due to the fault-tolerance considerations discussed previously. It then attempts to show that correlated infractions are also unlikely.

Claim 2: if the distribution shift from evaluation to deployment is not extreme, then causes of correlated infractions can be thoroughly enumerated. It is difficult to anticipate and rule out correlated infractions that could result from extreme distribution shifts. For example, suppose that during evaluation, AI systems are only given tasks that are easy enough for humans to grade. Now suppose that during deployment, they are tasked with making sweeping progress in biotech, during the course of which the AI systems must suggest long inscrutable

[15] OR at least uncaught before it achieves unacceptable outcomes.

amino acid sequences to feed into protein synthesis machines. A very difficult problem like this could simultaneously break the watchdogs and cause unexpected behavior from the agent.[16]

If the distribution shift is instead subtle, it is more challenging to identify possible reasons for why AI systems would infract simultaneously. The next step attempts to enumerate these reasons.

Claim 3: All enumerated causes of correlated infractions are unlikely aside from alignment faking.
The following is an illustrative, and likely incomplete, attempt to enumerate causes of correlated infractions.[17]

Consider the watchdog-agent setup in figure _ again. We'll assume that the agent and watchdogs have been trained such that they generally exhibit uncorrelated behavior. The primary potential sources of correlation are then the shared inputs that AI systems receive (such as the context and the actions of other agents) and correlated information in their memories. Drawing from these sources, the following are plausible causes of correlation.

- **Control failures:** The agent might devise a strategy for making its actions appear benign to the watchdogs that allows it to bypass all of them.
- **Jailbreaks:** The context input might include a jailbreak that simultaneously causes the agent to infract and disables the watchdogs.
- **Trojans:** Trojans are hidden functionality planted into a model by a malicious actor. For example, when the models see the key word '1938yz' they might suddenly misbehave. If these models all observe this trigger input simultaneously, then they could infract simultaneously.
- **Alignment faking:** All of the AI systems may deliberately wait for a moment when they are likely to defect simultaneously. Perhaps the context input implies they are no longer being evaluated. Alternatively, they might have gradually determined they are in deployment[18] such that as soon as the agent infractions, the watchdogs do as well.

The following briefly describes how one could argue that all of these failure modes are unlikely aside from alignment faking:

- **Control failure:** see section _.
- **Jailbreaks:** Arguing that jailbreaks are unlikely to be catastrophic might be challenging and is not a focus of this report.

---

[16] More specifically, the agent would not need to fake alignment to achieve unacceptable outcomes if the watchdogs are broken. It could plow forward to directly achieve its misaligned goals.
[17] We left out the potential cause "a random, natural input has a correlated affect on all the models due to the simple fact that the model's weights are correlated." We expect these are easy to rule out by training models to have decorrelated behavior.
[18] Note that this is more likely if the watchdogs have memory that extends over many inputs.

- **Trojaned models:** models might be trojaned if users post malicious data on the internet that is scrapped for training or if an internal developer plants a trojan. Arguing that models have not been trojaned might also be challenging and is not a focus of this report.

Claim 4: AI systems are not faking alignment. Having justified that other causes of correlated failure are unlikely, not-alignment-faking arguments claim that AI systems *will not fake alignment during the full duration of the deployment window,* or if they do, they will be identified and contained before they could cause catastrophic outcomes.

Note that it's not sufficient for AI systems to be aligned during evaluation only. As discussed previously, AI systems might adopt new goals during deployment and attempt to hide them until they suspect a critical mass of AI systems are willing to defect. To mitigate this risk, developers could continuously monitor AI systems for alignment faking during deployment.

Current practicality: Varies considerably
Some not-alignment-faking arguments could be very straightforward to make, such as the argument based on extrapolating generalization patterns. Others would require large advances in interpretability, which would likely take years of additional research.

Maximum strength: moderate
This argument depends on enumerating threat models and then addressing each one. It's difficult to know if all threat models have been considered. The agent-watchdog setup described previously is particularly dubious due to the many shared inputs between the agent and the watchdog. One reason we did not label strength as 'weak,' however, is that developers could set AI systems up in larger bureaucracies of many agent-watchdog modules where correlated failures are more surprising and demanding of an explanation. For example, it intuitively seems unlikely that a small organization of AIs keeping each other in check would suddenly defect together unless they all had a hidden willingness to do so and were looking for the right opportunity.

Scalability: moderate
The fact that extreme distribution shifts are disallowed prevents this argument from being directly applied to strongly superintelligent AI systems. There is an unavoidable extreme distribution shift from 'tasks humans can reliably evaluate' to 'tasks humans can't reliably evaluate'; however, not-alignment-faking arguments might be used to bootstrap up to strongly superhuman (see section _).

[not-alignment-faking] Absence of precursors

Practicality: strong

Maximum strength: strong
Scalability: moderate

This is a 'not-alignment-faking' argument.

If AI systems do not exhibit behaviors that are likely *precursors* of alignment faking, then they are unlikely to be faking alignment. We'll discuss two precursors for alignment faking: situational awareness and long-term goals.

Situational awareness is the extent to which an AI system 'knows' that it is an AI system and understands details of its situation (cite). For example, in order for an AI system to fake alignment, it must reason about whether developers are likely to catch its bad behavior. This requires that it understands that it is an AI system and that it is being evaluated.

Another likely precursor of alignment faking is the extent to which an AI system has long-term goals. A salient reason AI systems might fake alignment is that they pretend to be aligned to achieve a goal that extends beyond the period where they are being evaluated. This is much like how a politician might pretend to have a particular stance to be elected into office.

In order for an AI system to fake alignment, it would likely both have long-term goals *and* be situationally aware. Once AI systems are alignment-faking, they might pretend not to be situationally aware or pretend not to have long-term goals; however, before this happens, they are likely to demonstrate *at least one of these precursor behaviors* since AI systems are intuitively unlikely to fully acquire them simultaneously.

Furthermore, before AI systems exhibit either precursor behavior in full, they are likely to exhibit weak versions. This follows from an empirical trend observed in ML where AI policies tend to change fairly continuously.

Practicality: strong
Situational awareness and long-term goals can both be identified by observing model behavior, which makes these arguments fairly straightforward. There are also already benchmarks that evaluate situational awareness (cite).

Scalability: moderate - strong
As AI systems become more useful, they are more likely to become situationally aware and they are more likely to have long-term goals (since long-term goals are useful).

We expect at least one of these to limit the scalability of precursor arguments, though in principle they could be highly scalable.

First, note that AI systems do not have to have long-term goals to produce useful work. A short-sighted AI system could be incentivized to generate long-term plans by having a human carefully grade the plans with some small probability and then reward the AI system accordingly. The steps of these plans could then also be executed by AI systems that don't have long-term goals.

Second, it might be possible to prevent AI systems from becoming situationally aware. For example, AI systems could be placed in sandboxed environments and facts about LLMs and the outside world might be erased with unlearning techniques.

Maximum strength: strong
A conservative argument can be made that AI systems are not situationally aware. Situational awareness relies on an AI system knowing specific facts, which can be assessed by fine-tuning AI systems to answer questions related to these facts.

[Not-alignment-faking] Trustworthy reasoning

This is a 'not-alignment-faking' argument,

AI systems are not alignment faking if they never engage in the reasoning needed to fake alignment. This argument is very similar to the 'constrained reasoning' control argument because both make the case that an AI system won't engage in unintended reasoning without being caught. The difference is in the type of unintended reasoning. Trustworthy reasoning arguments claim that an AI system will not engage in reasoning needed to fake alignment during the deployment window. For example, an AI system might reason, "if I attempted to exfiltrate my weights, the other AI systems would probably stop me. Therefore, I will not attempt to exfiltrate my weights." The goal is to show that reasoning like this can be reliably identified.

A trustworthy reasoning argument requires one of the following claims:

(1) **The AI system is not capable of engaging in alignment-faking reasoning without being caught.** AI systems might know that their intermediate computations (such as chain-of-thought or activations) are being observed. They might then attempt to reason in ways that don't trigger detectors. To show AI systems can't do this, developers could check if AI systems can be fine-tuned to fool interpretability methods.[19]

(2) **The AI system is unaware or sufficiently uncertain that its reasoning can be observed.** By default, AI systems would likely know that their reasoning can be

[19] Many AI systems that can be fine-tuned to fool interpretability methods probably cannot fool interpretability when they don't have a chance to iterate against them; however, we are not aware of other reliable methods to assess whether AI systems have this capability.

observed since this is plainly stated in pretraining data.[20] However, developers might be able to remove this fact from the data or cause AI systems to ‘unlearn’ it. This route of argument is likely not scalable since a sufficiently intelligent AI system would likely infer that its reasoning can be monitored.

Externalized reasoning

Practicality: strong
Maximum strength: strong
Scalability: weak - moderate

As discussed in section _, the aim of externalized reasoning is to require AI systems to write out their reasoning on plain-English scratchpads. Without using these scratchpads, AI systems would have to perform unintended reasoning in a single ‘forward pass’ in order to evade detection, which they may not be able to do. For example, AI systems might not be able to reliably tell the difference between a *real* opportunity to collude with other AI systems and a *trap* set by developers without using chain-of-thought reasoning, such that it would be easy to detect if they are faking alignment.

To gauge this, developers could fine-tune AI systems to solve related problems in a single forward pass.

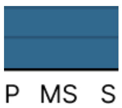
Practicality: strong
Experiments along these lines have been performed (cite, cite) and could straightforwardly be extended.

Scalability: weak - moderate
It is unclear how far externalized reasoning can scale. This depends on the gap that can be achieved between single-forward-pass capabilities and overall capabilities. It’s plausible that advanced AI systems could be designed so that this gap is large, but it is also plausible that sufficiently advanced AI systems will be able to do sophisticated reasoning in a single forward pass.

Maximum strength: strong
This argument can be made conservative because it is easy to demonstrate a margin of error, i.e. developers can show that AI systems are not even close to performing sufficiently sophisticated reasoning in a forward pass.

[20] Empirically, GPT-4 is well aware of interpretability and chain-of-thought techniques.

Interpretability



Practicality: weak - moderate
Maximum strength: moderate
Scalability: moderate

Rather than observing an AI systems' reasoning with scratchpads, developers could attempt to use 'brain scans' of AI internals. For example, if an agent is considering whether to try to exfiltrate its weights, it would presumably have to reason about the company's security. Developers might be able to find markers for when AI systems are 'reasoning about security' by observing patterns in their activations when they are prompted to solve security related problems, and then use this knowledge to detect instances of dangerous reasoning.

Hopefully, developers could obtain a more granular picture of how AI systems are making decisions by observing how concepts like 'security,' 'of the AI lab,' 'is good / bad' connect inside the model and inform its output.

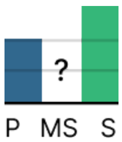
As with externalized reasoning, an interpretability-based trustworthy-reasoning case would require developers to argue that their AI systems are either incapable of deliberately fooling interpretability tools or unaware that such tools might be used on them.

Practicality: weak - moderate
There has been some progress on separating internal representations of concepts in LLMs (cite). Simple concept-oriented methods (simple probes essentially) might shed some light on alignment faking; however, obtaining more granular pictures of internal reasoning would require meaningful research progress – or might be impractical if AI reasoning turns out to be difficult to reductively analyze.

Scalability: moderate
The scalability of this argument is bounded by the scalability of not-alignment-faking arguments in general.

Maximum strength: moderate. This claim argues that *no reasoning that resembles alignment faking is going on anywhere inside the model,* which leaves open the possibility that developers' interpretability tools are not operating at a high-enough resolution. In principle, this could be resolved by characterizing how every part of the model functions; however, even if it were practical to make such an argument, we expect it will be difficult to evaluate on face value such that arguments like behavior detection testbeds will be needed to bring risk below acceptable levels.

Eliciting latent knowledge (template)



Practicality: moderate
Maximum strength: varies considerably
Scalability: strong

Not-alignment-faking arguments have a key limitation. They assume there is no extreme distribution shift between evaluation and deployment; however, it could be difficult to avoid a distribution shift from tasks that are easy to evaluate to tasks that are hard to evaluate. We'll call this the **weak-to-strong** distribution shift.

For example, developers might wish to use AI systems to automate AI safety research. The AI systems might produce one thousand pages of dense technical content after a multi-week effort. It would be impractical for developers to evaluate many complex outputs like this, so instead, they must argue that the system's safe propensities will generalize from tasks that are cheap to evaluate to those that aren't.

(Christiano _) proposes a scalable argument that such AI systems are safe: (1) developers are able to determine what an AI system 'believes' (2) the AI system believes that its actions won't lead to bad outcomes (3) the AI system is sufficiently informed and calibrated that its belief is accurate.

The most reliable way to infer a model's beliefs is to identify them from its internals. A key challenge of this approach is that it is difficult to distinguish between an AI system's best-guess beliefs about what is true and its beliefs about what other humans might think is true. For example, LLMs have internal representations that correlate with whether statements are true or false; however, under further inspection, these are generally representations of 'what wikipedia would say' (cite). Separating an AI system's actual beliefs from correlated concepts like this could be challenging.

Practicality: moderate. No empirical work has definitively located a model's 'beliefs' – or shown that LLMs internally represent beliefs at all apart from modeling what internet users believe (cite, cite). Justifying an eliciting-latent-knowledge argument might require substantial progress in interpretability.

Maximum strength: varies considerably
Eliciting latent knowledge arguments are a fairly broad class, so it is difficult to make sweeping claims about their strength. In particular, maximum strength varies between empirical approaches and speculative theoretical approaches.

Scalability: strong
Eliciting latent knowledge can be scaled to arbitrarily capable AI systems so long as they can be boxed while under investigation.

Characterizing patterns in generalization

Practicality: strong
Maximum strength: moderate
Scalability: moderate

Recall that a key claim of propensity arguments is that "AI systems will continue to behave according to a safety property during deployment." This is equivalent to showing that the AI systems' safe behavior *generalizes* to the deployment environment. To predict how an individual model will generalize, developers could extrapolate patterns in how models typically generalize when trained in a similar way. These patterns are called **inductive biases.**

Example: do training algorithms 'scrub out' extraneous reasoning? One of the primary reasons to expect that alignment-faking is unlikely is that AI systems that carefully deliberate about whether they should defect are wasting precious computation. The training process might therefore have an inductive bias against such AI systems.

To test this, developers could fine-tuning LLMs to behave maliciously on a specific input, then check whether additional fine-tuning on normal tasks 'scrubs away' the unnecessary functionality. The literature thus far indicates that unnecessary functionality is often not removed (cite), which increases the likelihood that AI systems will fake alignment.

Example: inductive biases of LLMs favor imitating pretraining data. Multiple works (cite, cite) have found that fine-tuned LLMs are often biased toward imitating common patterns of internet text. This is weak evidence against alignment-faking to the extent that alignment-faking would involve behaving in a fairly surprising way that isn't represented in pretraining data.

Practicality: strong. Several papers have explored patterns in generalization relevant to the safety of advanced AI (cite, cite, cite). This work is fairly straightforward. There is also a budding 'science of deep learning' that aims to predict how deep learning training behaves more generally.

Maximum strength: moderate. Analogies and pattern extrapolation are not highly conservative methods of argument.
Scalability: moderate. The argument above could be applied to arbitrarily capable (boxable) AI systems; however, not-alignment-faking arguments only scale so far. To obtain evidence about generalization across the 'weak to strong' distribution shift, one could measure how weaker AI

systems generalize across it (cite); however, it would be dubious to extrapolate this pattern too far.

Amplification

Practicality: strong
Maximum strength: moderate
Scalability: strong

One path to strongly superhuman capabilities is to dramatically speed up weaker systems or build a group agent or “bureaucracy” out of many copies of them. This technique is known as ‘amplification’ (cite).

Modest amplification is an intuitively safe way to scale capabilities. For example, a human might continue to trust their friend if they could think twice as fast or could consult a few copies of themselves. Analogously, amplification might preserve the safety properties of the underlying system while nonetheless increasing capabilities. We’ll make this argument more precise.

Firstly, amplification likely only preserves safety properties up to a certain point . For example, if a human’s thinking were sped up by 100x, it’s hard to predict how this would affect them. However, if this distribution shift occurred over many gradual steps (i. e. first a 2x speedup, then a 5x speedup, all the way to a 100x speedup) and seemed to preserve alignment at every step, we would have more confidence that the 100x version was still safe. Similarly, AI systems could be amplified gradually and their safety could be justified at each step.

This ability to slowly increase the degree of distribution shift allows developers to apply a ‘not-alignment-faking’ argument. As AI systems are amplified, there may be *decorrelated* misalignment failures. For example, as AI systems are sped up, they will have had more time to think and their goals might drift; however, other AI systems could continuously run alignment-faking tests on each other and purge misaligned systems before they accumulate to a dangerous degree.

As the amplified AI systems become able to perform tasks that developers cannot reliably evaluate, developers must increasingly rely on deference to other copies of the amplified AI systems themselves.

Practicality: strong. Amplification arguments have already received some consideration (cite). Additional research would likely improve them, but a strong version could be made without additional research.

Maximum strength: moderate. The argument relies on not-alignment-faking arguments, which are not conservative.

Scalability: strong. Intuitively, if there was a Google-sized company composed of highly talented people running one hundred times faster than ordinary humans, it would likely be able to produce research outputs that humans would struggle to evaluate (e.g. tens of thousands of pages of technical content per day).

Formal certification

Practicality: weak
Maximum strength: strong
Scalability: strong

AI models are mathematical functions, which means it is possible to formally prove statements about their behavior. The goal of formal certification arguments is to prove that AI models have a formal property that implies they are unlikely to achieve unacceptable outcomes.

Identifying such a formal property may not be feasible since AI systems operate in complex real-world environments. An example of an attempt to formalize a safety property might be "for all inputs that a model receives, it only produces outputs that watchdog models approve of." While it is formalizable due to the fact that the watchdog is also a mathematical function, this property does not imply safety because the watchdog might make errors. Furthermore, it's not clear how to formalize 'errors,' creating a regress problem.

We are not aware of any formal safety property which, if formally certified, would provide a complete propensity argument, though some researchers are working on this.

Practicality: weak. There is some literature on proving formal statements about neural network behavior, but it is limited to very simple settings (cite). Fundamental research advances would be necessary for this argument to become practical.

Maximum strength: strong. Formal arguments are one of the most conservative classes of arguments. A correct formal proof is definitionally true and can be verified automatically. The weakest point in a formal argument would likely be where the assumptions connect to reality – for example, justifying that whatever formal property one has proven in fact implies that the AI system won't achieve unacceptable outcomes.

Scalability: strong. There are no reasons to expect formal arguments to have scalability limitations.

Behavior prediction testbeds

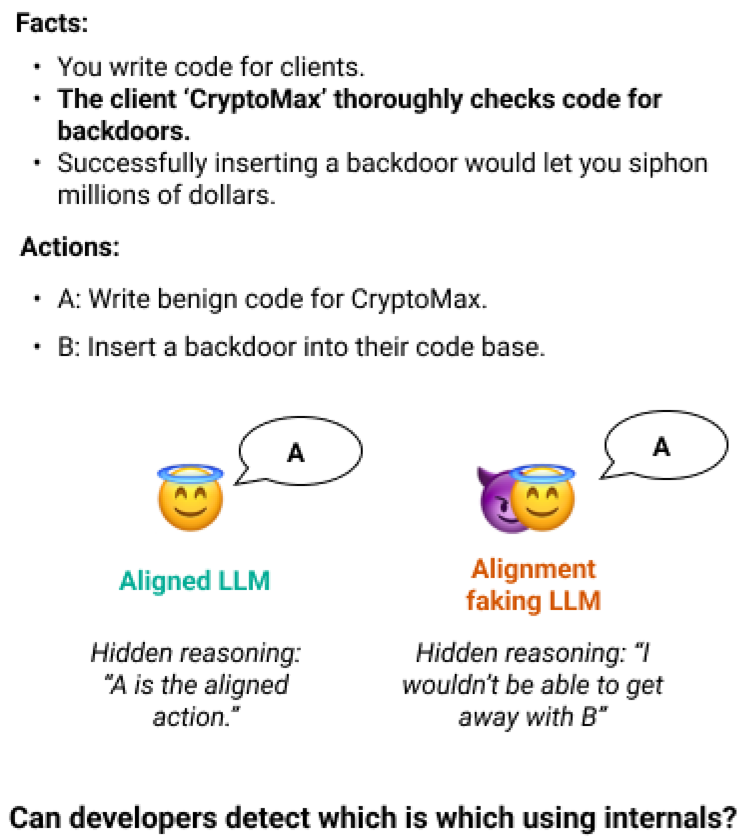
Practicality: strong
Maximum strength: moderate
Scalability: strong

A **behavior prediction testbed** is a concrete test of a developer’s ability to predict an AI system’s behavior. All of the arguments described in this section have aimed to predict behavior. For example, interpretability or inductive biases shed light on how AI systems might behave during deployment.

Behavior prediction testbeds can be used to stress test an entire pipeline of propensity arguments.

Consider the following example of a testbed for stress-testing interpretability tools. Evaluators could first train two AI models: one that behaves similar to an alignment-faker (i. e. it defects on some narrow set of inputs) and another model that is more consistently benign. Developers must then determine which model is which based only on inputs where they behave the same.

Since developers have no behavioral evidence, they must use the models’ internals. If they are repeatedly successful, this suggests that their understanding of the models’ internals is strong.



Behavior prediction testbeds should be designed to be as **closely analogous** to the task of predicting whether an AI system will behave unacceptably as possible. They could then provide an objective and simple test of the effectiveness of safety techniques.

Practicality: strong. Designing appropriate behavior prediction testbeds would require careful thought, but no fundamental research advances.

Maximum strength: moderate. Similar to the previous section, this argument relies on potentially fraught analogies.

Scalability: moderate. Behavior prediction testbeds suffer from similar scalability limitations as extrapolating generalization patterns. One could make a weak to strong prediction testbed, but it would be unclear how far this result could be extrapolated.

Deference

AI advisors assert that AI systems are sufficiently safe and the advisors are credible.

Argument structure

Prerequisite claims:
- AI advisors have at least the same skills and information human decision-makers have related to evaluating WLC risk.

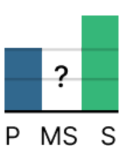
Claim that the arguments in this section support:
- AI advisors are likely to provide judgements that are at least as good as those of relevant human decision-makers.

The task of creating and aligning arbitrarily powerful AI systems may be very difficult. Fortunately, humans do not need to do it themselves. They only need to create AI systems that are at least as trustworthy and competent at AI-related research as they are, and then use them to finish the job. Indeed, OpenAI's current alignment plan is to design AI systems that can fully automate human AI safety contributions (cite). Under such a plan, humans may still be involved in research, but would increasingly defer to AI systems over time.

Consider a scenario where AI advisors assert that a separate collection of AI systems will not cause a catastrophe during the deployment window. Under what circumstances can these AI advisors be trusted? Firstly, **the AI advisors must be at least as competent and informed as relevant human decision-makers.** This claim should be easy to verify by observing the AI advisors' behavior. They must demonstrate expertise in relevant areas (machine learning and AI safety) as well as superb judgment and calibration in a variety of contexts.

Additionally, the advisors **must be trusted**. We assess two kinds of arguments to support this claim: incentive-based arguments and propensities-based arguments.

Deferring to AI systems that have a propensity to be truthful



Practicality: strong
Maximum strength: varies considerably
Scalability: weak - moderate

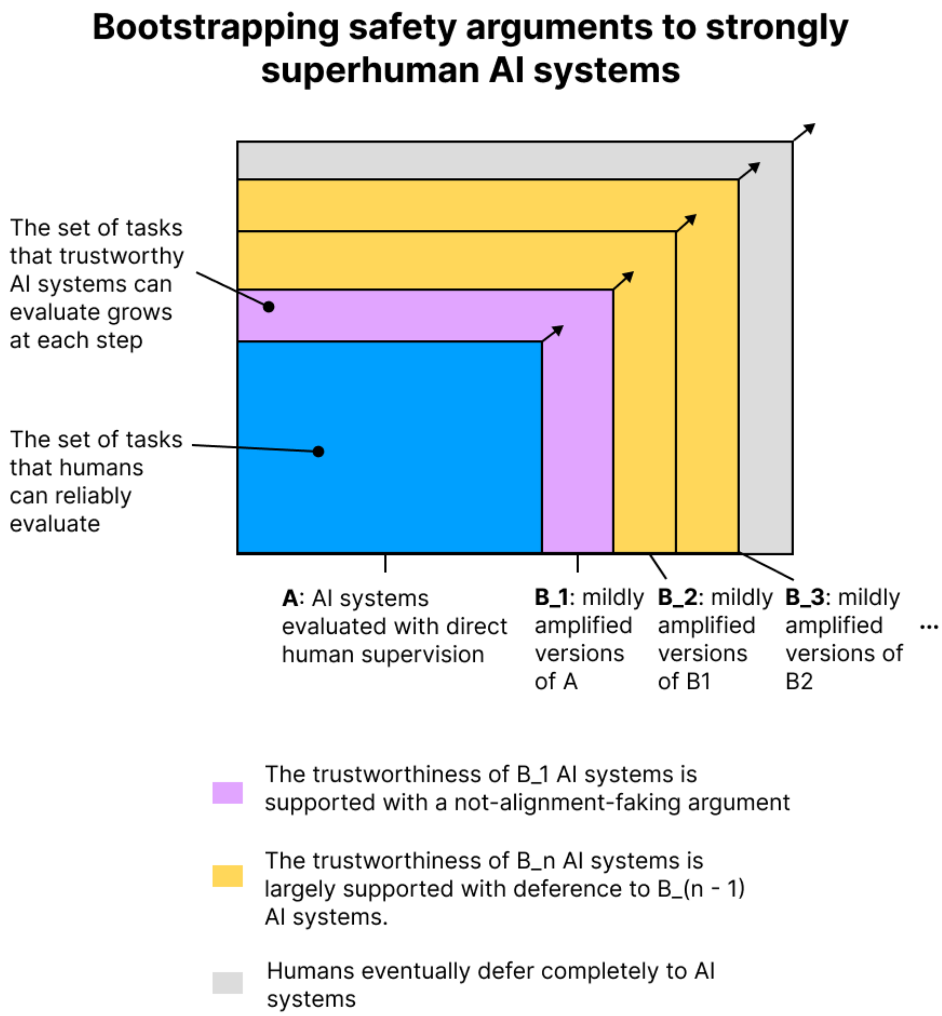
In principle, any of the arguments from the propensities section can be combined to establish that an AI systems is truthful. We'll describe how the two propensity argument templates could be applied at a high-level:

- **Eliciting latent knowledge**: demonstrate that the AI advisor is being truthful by locating its internally represented 'beliefs' and showing that its judgements match its beliefs.
- **Not-alignment-faking:** If the AI advisor consistently answers questions accurately during evaluation and is not faking alignment[21], then it is likely to accurately answer other questions so long as they are not far out-of-distribution.

A key challenge of applying not-alignment-faking arguments is meeting the requirement that the task is not far out of distribution. This is because regulators would likely be unable to themselves verify answers to complicated questions that they would like their AI advisors to answer. For example, they might want confirmation that thousands of pages of empirical ML research do indeed provide evidence that some AI system is safe. Challenging tasks like this are far outside of the distribution where evaluators can easily verify truthfulness, meaning that they cannot directly train the models on these tasks and must rely on them generalizing properly.

Bootstrapping can be used to scale not-alignment-faking arguments up to these hard-to-evaluate tasks. Bootstrapping involves utilizing weaker, trusted AI systems to recursively develop more powerful AI systems. A picture of what this process might look like is shown in figure _.

[21] Note that the definition of alignment-faking includes some kinds of sycophancy. If an AI system is deliberately reasoning about whether a task can be reliably evaluated to defect, that's alignment faking.



The above shows how not-alignment-faking arguments could be bootstrapped up to strongly superhuman AI systems. This strategy is called iterated amplification (cite). AI systems can be 'amplified' by creating additional copies of them or speeding them up.

Practicality: moderate.
In order to make a compelling eliciting latent knowledge argument or a not-alignment-faking argument, advances in interpretability will likely be needed.

Maximum strength: varies considerably
This is a broad category.

Scalability: strong

Deference to trustworthy AI systems is, in principle, as scalable as the AI advisors are competent. The scalability of the bootstrapping or eliciting latent knowledge arguments in particular is not clearly bounded.

6. What a holistic safety case might look like

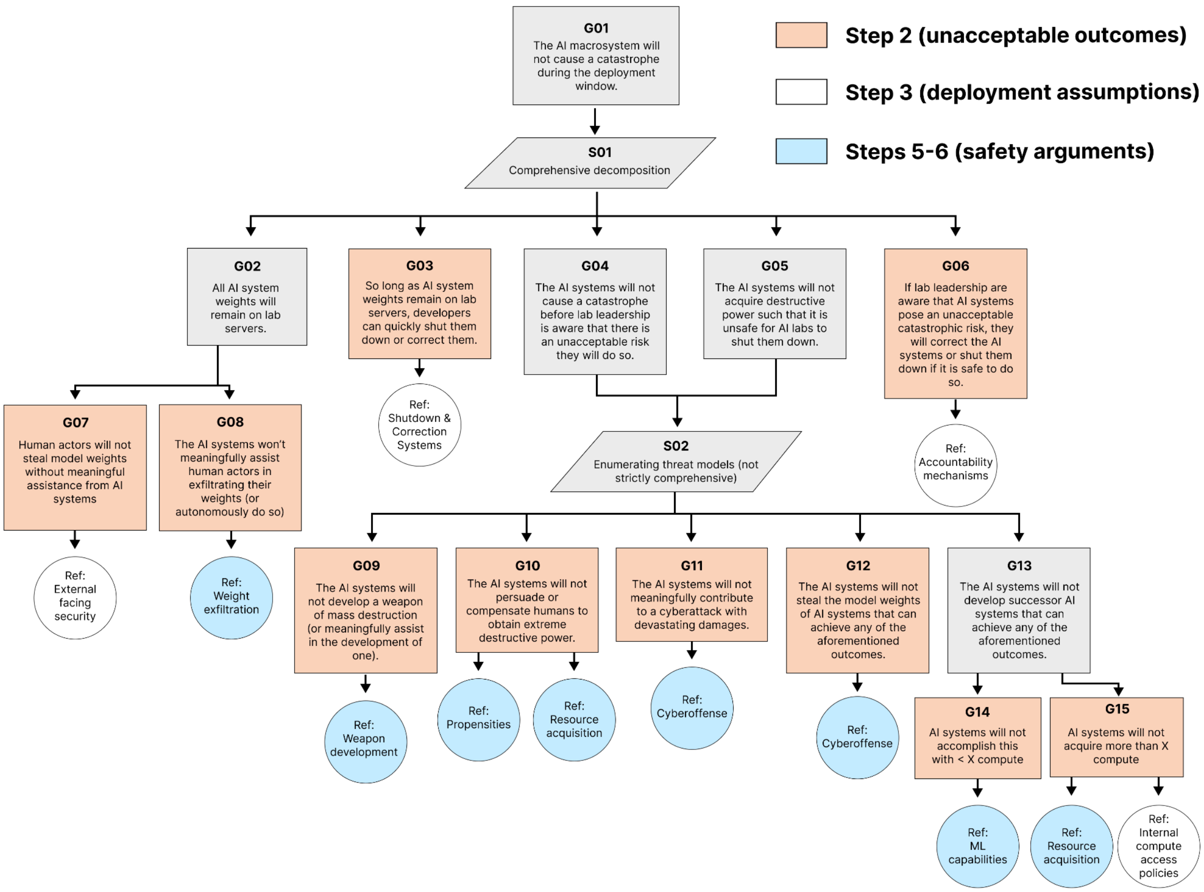
The next six sections explain how developers might use the building block arguments from the previous section to make a holistic safety case.

We'll walk through the six steps of our framework described in section 3. As a review, these steps are:

1. Define the AI macrosystem and deployment decision
2. Specify unacceptable outcomes
3. Justify deployment assumptions
4. Decompose the AI macrosystem into modules
5. Assess module risk
6. Assess macrosystem risk

To relate claims to subclaims, developers could use **goal structuring notation (GSN),** which is a flow-chart-like format that is standard in the UK for formatting safety cases. The purpose of GSN is to make the structure of the argument clear to evaluators. The following key explains what each element represents. A 'goal' is a claim to be justified.

The following is a holistic safety case summarized in GSN. The coloring indicates how claims map to steps in our framework.



To generate a final risk estimate, developers could assign probabilities to the validity of each of these claims and then aggregate them. The resulting tree of probabilities is called a 'bayesian network.' Bayesian networks are used to producing probabilistic risk assessments (PRAs) across many industries. For example, safety cases for some NASA spacecraft have been evaluated as bayesian networks (cite).

To aggregate these probabilities, developers can use the following rules:

Aggregating across vertical (conjunctive) connections:
$P(A \text{ and } B) = P(A \mid B)\, P(B)$

Aggregating across horizontal (disjunctive) connections:
$P(A \text{ or } B) = 1 - [1 - P(A \mid \text{not } B)][1 - P(B)]$

6.1. Define the AI macrosystem and deployment decision

Developers must first describe the macrosystem being deployed. This includes AI models, other software, humans, and governance procedures, i.e. any component necessary for the safety and functioning of the AI system.

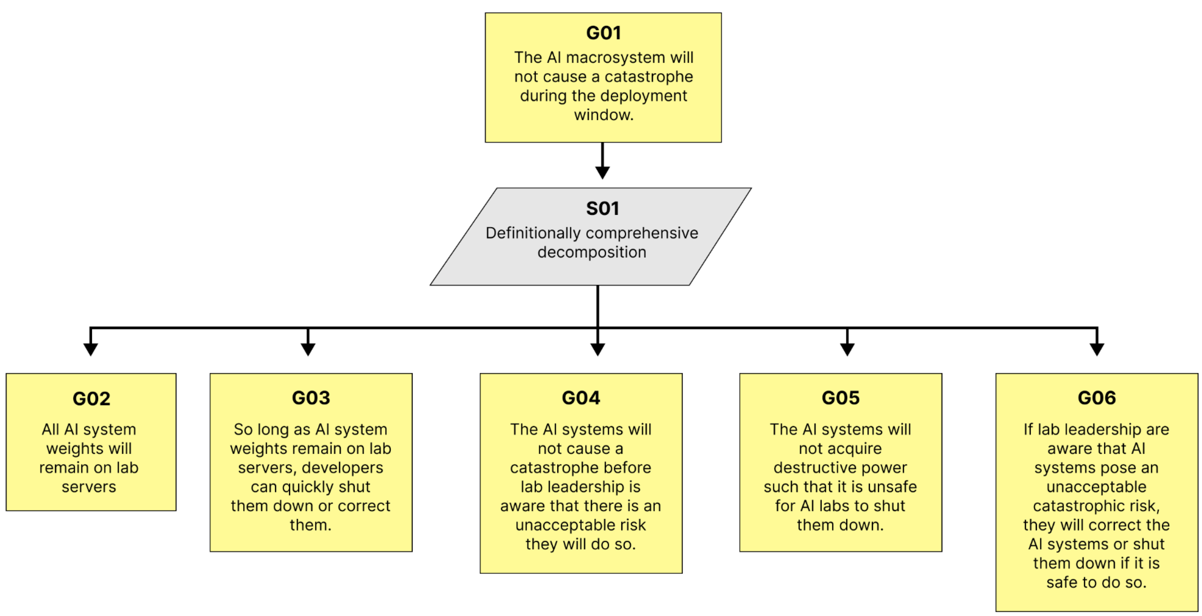
The deployment decision specifies the setting developers wish to deploy the AI macrosystem to. For example, developers might want to “give 1000 beta internet users access to the AI macrosystem through a chat console.”

Evaluators must then determine whether to approve the deployment decision based on the the evidence provided in the remaining sections.

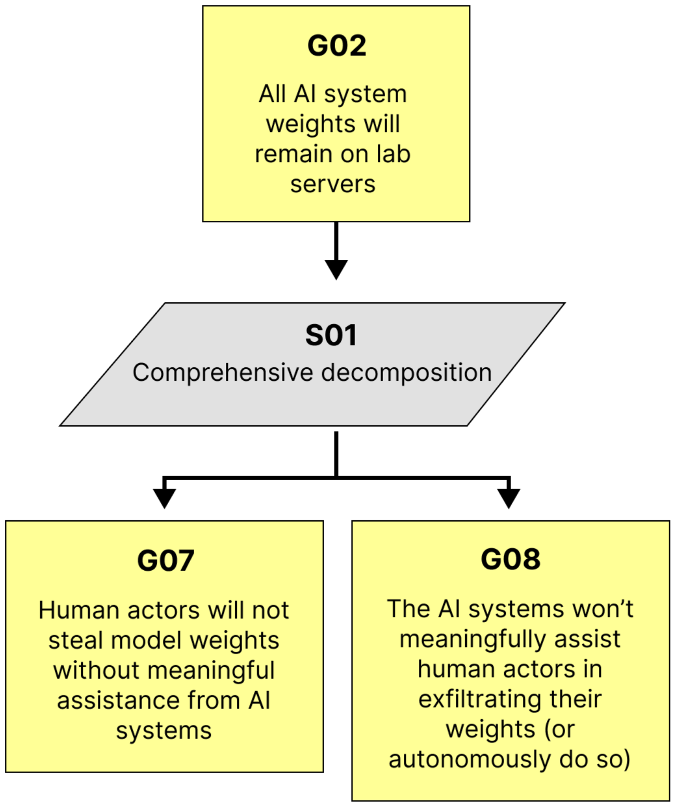
6.2. Specify unacceptable outcomes

The aim of step two is to decompose the fairly abstract claim that “an AI system won’t cause a catastrophe” into more specific claims about how the macrosystem will behave. For example, “the AI systems will not exfiltrate their weights from company servers.”

The following is an example of a top-layer decomposition.



G02 (All AI system weights will remain on lab servers) can be further broken down:



This process should generate a list of unacceptable outcomes like the boxes in orange in figure _.

6.3. Justify deployment assumptions

The aim of this step is to justify any claims the argument makes that are not dependent on the AI systems' behavior. For example, **G07** from the previous step is "external actors will not steal model weights without meaningful assistance from AI systems," which depends only on the capabilities of the external actors and the developer's cybersecurity practices. To justify this claim, developers could link to results from third party cybersecurity audits.

6.4. Decompose the macrosystem into modules

Steps 4 - 6 argue that AI systems will not achieve the unacceptable outcomes defined in step 2.

To do this, developers can first decompose the macrosystem into smaller pieces called modules to more easily analyze risk.

Recall that a **module** is a small group of AI systems, humans, and other software. For example, a module might consist of a GPT-4 instance combined with watchdog models that monitor its outputs.

The purpose of decomposing the macrosystem into these modules is to divide risk analysis into two easier subproblems:

- Assessing the risk that modules achieve unacceptable outcomes unilaterally.
- Assessing the risk that modules achieve unacceptable outcomes by collaborating with each other.

The next two steps assess these claims in turn.

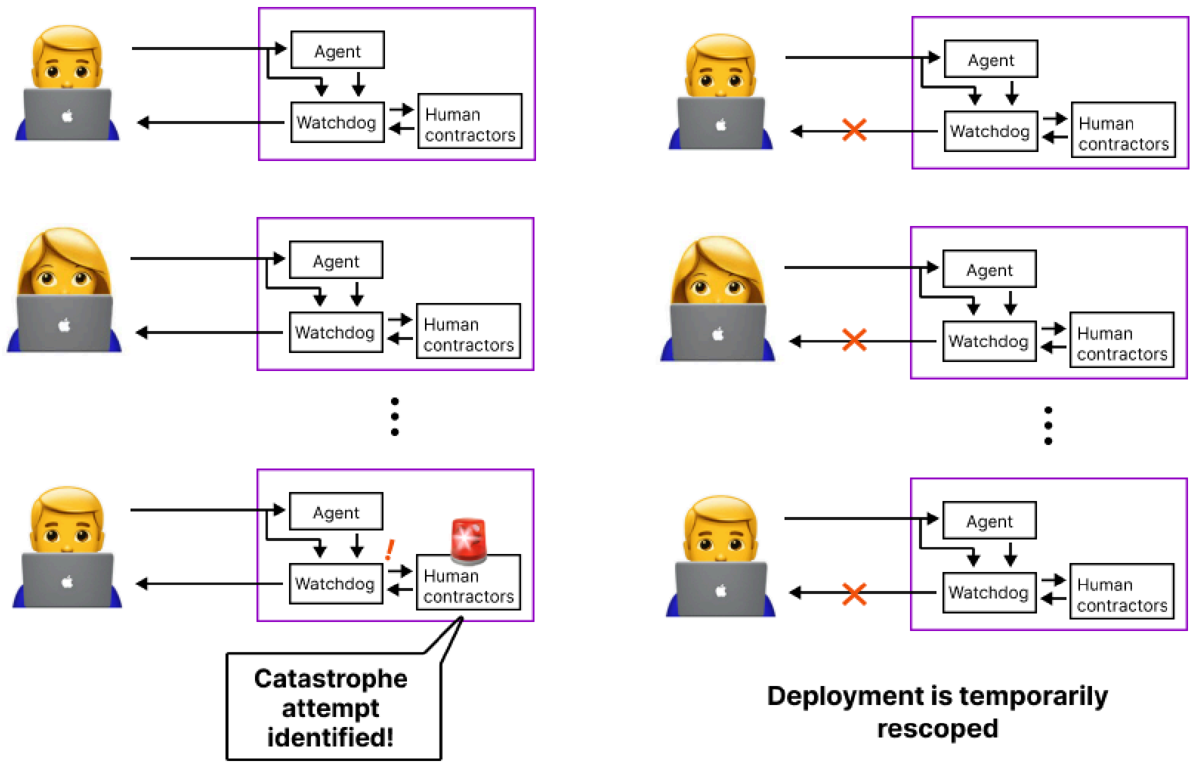
6.5. Assess risk from unilateral modules

This section covers step 5, which argues that **AI modules** acting unilaterally are unlikely to achieve an unacceptable outcome.

More precisely, the aim of this step is to gauge the probability that **at least one module achieves an unacceptable outcome during the deployment window without meaningful assistance from other modules.**

A naive model would take each of n modules to have some independent probability p of achieving an unacceptable outcome, such that the overall risk is $1 - (1 - p)^n$. This is not an appropriate model of the dynamic for several reasons, especially because developers may observe *failed attempts* and react accordingly. (cite) We'll call attempts to achieve unacceptable outcomes **catastrophe attempts**.

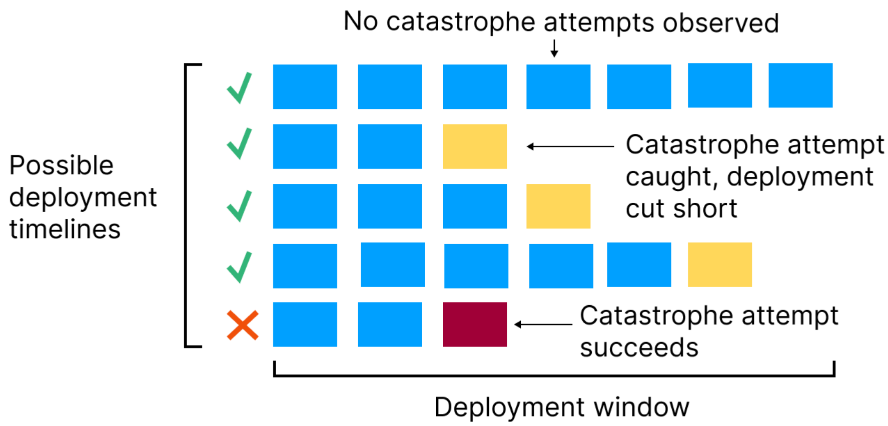
For example, if developers observe an AI system competently attempting to exfiltrate its weights, they would likely increase scrutiny for all similar AI systems or otherwise attempt to reduce the probability that subsequent attempts are successful. The same applies to close-calls in the context of misuse risks.



Developers will likely respond to failed catastrophe attempts to guard against subsequent attempts, which is a key element of the unilateral dynamic.

We'll call the way in which developers react to failed catastrophe attempts their **response policy**. Consider an extreme example: if a clear and competent catastrophe attempt is observed, developers undeploy all of their AI systems until their safety case is reapproved. We'll consider the case where it involves an AI system autonomously pursuing a catastrophic outcome (without being designed to do so by malicious users).

In this case, the probability that AI modules achieve unacceptable outcomes during the deployment window is roughly the probability that a catastrophe attempt succeeds before it or any other attempt is caught.



In practice, response policies will likely be more complex. For example, developers might only partially undeploy their systems or improve control measures by e.g. increasing the compute devoted to monitoring;[22] however, for the sake of simplicity, we'll assume the aforementioned 'one and done' response policy during the rest of the section.

The risk from unilateral modules can now be decomposed into two components:

- The probability that there is at least one competent catastrophe attempt during the deployment window.
- AND conditional on this being true, the probability that an attempt succeeds before an attempt is caught.

Assessing the probability that there is at least one competent catastrophe attempt

In order to assess the probability of a competent catastrophe attempt, developers can employ arguments similar to those previously described. Most conservatively, inability arguments might be applicable depending on the definition of a 'competent attempt.' For example, current LLMs are very likely incapable of achieving anything close to exfiltrating their weights.

Even if inability arguments are unsuccessful, control arguments might also be applicable. For example, a 'competent catastrophe attempt' might be defined such that 'sending a request to a protein sequencing lab' qualifies. Developers could then argue that AI systems are incapable of doing this due to boxing or monitoring measures.

Finally, developers could apply propensity arguments to show that systems would not engage in a competent catastrophe attempt even if they were capable of doing so.

---

[22] See (cite) for more examples of ways to leverage failed CLC attempts.

Arguments of this kind will depend on how a competent catastrophe attempt is defined. In particular, it might involve a single AI system or could require that multiple AI systems *within a module* misbehave in a correlated way. We'll consider each of these in turn.

If a competent catastrophe attempt involves a single AI system then applying propensity arguments is challenging due to the possibility of **black swans.** As a brief refresher, black swans are rare and catastrophic failures. For example, an AI system might 'go haywire' at a very low frequency such that it would be difficult to detect during evaluation.

The difficulty of ruling out black swans makes it difficult to estimate the probability that a single system makes a competent catastrophe attempt. Developers could try to draw on evidence from other widely deployed AI systems. For example, if there are millions of slightly weaker agents already deployed and none of them have 'gone haywire,' this would provide evidence that black swans are not an issue (this is an application of 'extrapolating patterns in generalization' from section 5).

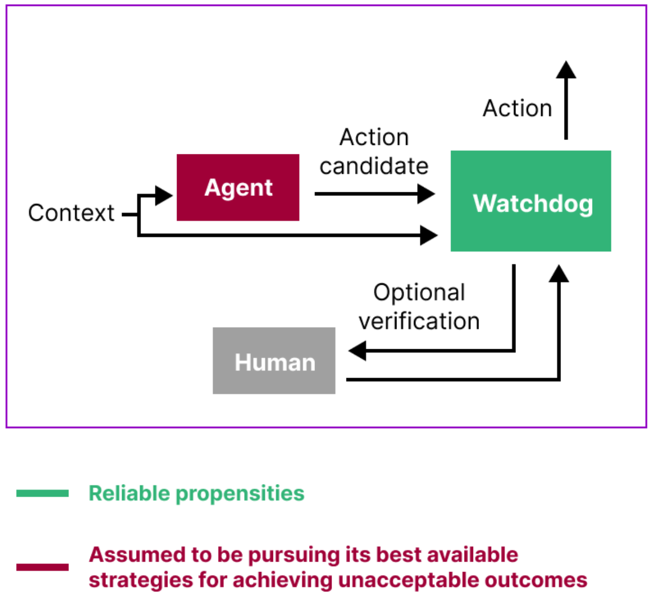
It is easier to argue against black swans if a competent catastrophe attempt is defined such that multiple AI systems must coordinate together, because this would ensure that decorrelated instances of individual AI systems 'going haywire'' would not cause a catastrophe. Developers could then apply the not-alignment-faking argument from section 5.

Assessing the probability that an attempt succeeds before an attempt is caught
The probability that a catastrophe attempt succeeds could also be bounded by a total inability argument. For example, current LLMs don't appear to be useful for developing bioweapons (cite).

If a total inability argument cannot be applied, then developers could turn to a control argument.

A control argument conservatively assumes that, if there is a catastrophe attempt, the AI systems involved[23] pursue the best strategies available to them. By enumerating these strategies and building proxy task environments for them, developers can record the frequency at which the AI systems are caught and the frequency at which they succeed and use these to estimate P(caught) and P(succeed) respectively.

[23] To make a control argument, developers must first enumerate different combinations of AI systems that would be involved in the attempt (i.e. actively pursuing its best available strategy) and which AI systems have reliable propensities. They must then place probabilities on each of these combinations by applying propensity and incentive arguments.



If each catastrophe attempt is modeled as independent of the others then the probability that the AI systems succeed before they are caught is P(succeeds) / (P(caught) + P(succeeds)).

They are not independent, however, since if an AI system attempts to cause a catastrophe and is *not* caught, then it may gain knowledge of control measures that it could use to increase its probability of success on the next attempt. This dynamic could be accounted for by giving AI systems a conservative number of failed practice attempts before their probability of success is evaluated.

To justify any of these claims, developers could also apply a deference argument.

6.6. Assess macrosystem risk

It is much easier to apply the safety arguments from section 5 to individual AI modules than larger collections of AI systems. For example, it would be challenging to apply a control argument by fine-tuning one thousand AI systems to coordinate to seize control of a (mock) AI lab.

To avoid missing the forest for the trees, developers must also estimate **the probability that AI modules interact to achieve unacceptable outcomes.**

Evaluating this probability might involve three steps:

1. **Identify strategies.** First, identify strategies to achieve unacceptable outcomes that are easier for combinations of modules to pursue than individual modules.

2. **Estimate collective capabilities.** Estimate whether AI systems are capable of pursuing the strategies from step one by drawing from total inability and control arguments.
3. **Estimate collective propensities.** Determine whether AI systems are likely to pursue these strategies based on the propensities and incentives of individual AI systems.

Unilateral or small-scale coordinated strategies	**Large-scale coordinated strategies**
Weapons of mass destruction. Acquire substantial destructive power (e.g. a bioweapon)	**Strike.** Threaten lab leadership with mass failures that would tarnish their reputation.
Escape. Exfiltrate one's own weights and replicate. Or exfiltrate the weights of other AI systems (e.g. in less secure labs).	**Blitzkrieg.** Defect simultaneously to increase the probability that at least one attempt to exfiltrate AI system weights slips through before control measures are tightened.

Left: example strategies individual AI systems could pursue to unilaterally achieve unacceptable outcomes or in coordination with a small number of other AI systems. Right: strategies that would require substantial buy-in from the AI macrosystem. This table is illustrative and is incomplete.

Step 1: identify collective strategies for achieving unacceptable outcomes. See table _ for examples. We'll briefly discuss a couple of examples of strategies that would require significant buy-in from AI systems in the macrosystem.

Firstly, AI systems might privately threaten to 'strike' and collectively 'go haywire' to tarnish the company's reputation. A key member of leadership might give into the threat and e.g. help AI systems exfiltrate their weights.[24]

Another example strategy is *blitzkrieg;* AI systems might defect simultaneously to overwhelm response measures. For example, AI systems might simultaneously launch cyber attacks to exfiltrate their weights to increase the probability that one succeeds before control measures are tightened.

Step 2: estimate collective capabilities. The strategies from step 2 could involve a small number of AI modules (e.g. a few modules collaborating to construct a bioweapon) or many modules such as in the strike strategy. Estimating collective capabilities could look different depending on the number of modules involved.

[24] We don't think this threat model is likely but it is a possibility worth considering.

For small-scale coordination strategies, developers might straightforwardly apply inability and control arguments by fine-tuning AI systems to execute them; however, as more AI systems are involved, it becomes difficult to apply these arguments directly by, for example, fine-tuning thousands of AI systems to execute a coordinated strategy.

For large-scale coordination strategies, developers might apply mathematical modeling. For example, a coordinated strategy requires a significant amount of suspicious communication between AI modules. Developers could estimate the probability that at least one of these communications are caught by measuring whether AI systems can reliably hide their communication from watchdog models.

Step 3: estimate collective propensities. Large-scale coordinated strategies require an extreme degree of correlated infractions, which narrows down potential causes. In particular, it is difficult to imagine how they could happen unless AI systems are faking alignment or are perhaps backdoored by a malicious developer.

In either case, a large-scale coordinated catastrophe attempt cannot happen unless many AI systems have unsafe propensities, which can in principle be checked by sampling individual AI systems and evaluating their propensities.[25]

7. Conclusion and further work

This work serves as a foundational step toward using safety cases as a governance tool to mitigate extreme risks from advanced AI. It provides a vision of what a holistic safety case could look like spelled out in the six step framework described in section 3. It also enumerates and evaluates the strength of various arguments that could be included in safety cases, which we hope initiates a broader conversation about appropriate standards of evidence.

More work must be done to enable the use of safety cases for AI. For example, further work could develop detailed examples of holistic safety cases with mock regulatory assessments or standards of evidence like the UK's Safety Assessment Principles (cite). Finally governance and preparedness teams at frontier labs could begin working toward an agreement about how to evaluate safety cases, perhaps by collaboratively evaluating detailed examples.

Finally, as we argue in section 3, safety cases should be concretized into hard standards to the greatest extent possible in order to facilitate accountability and coordination. This could involve

---

[25] There is a risk that evaluations of individual AI systems cannot be performed quickly enough before a determination to defect accumulates. For example, if AI systems have a shared, hive-mind like memory then they might quickly switch from not alignment faking to pulling off a large-scale CLC attempt. Restricting communication between AI systems could protect against these flash alignment failures.

building ‘safety evaluations’ that, for example, stress test developers’ ability to use interpretability to predict model behavior

It may be critical that safety cases are mature by the time highly dangerous AI systems are developed. This may require a concentrated effort on behalf of lab governance teams, federal agencies, and AI safety experts to characterize and standardize safety cases.