Why frontier AI safety cases need independent external review

Authors: Stephen Barrett[1], Francisco Javier Campos Zabala[1], Sean P. Fillingham[1], Umair Siddique[1], James Walpole[1], Robin Bloomfield[1,2], Henry Papadatos[3]

[1]Arcadia Impact AI Governance Taskforce
[2]City St George's, University of London
[3]SaferAI

April 2026

Summary

- **External review of frontier AI safety cases is valuable.** It helps counter the developer's own confirmation bias and conflicted incentives.
- **We found evidence of that value in practice.** In our review of Google DeepMind's public scheming inability safety case, we found both strong features and important gaps.
- **A good review is more than checking evaluation results.** It should examine the claim being made, the system and deployment context, the risk pathways to harm, the logic of the argument, and the strength of the evidence.
- **Reviewers need more than a paper.** AI developers should provide system models, risk analyses, evaluation details, and supporting safety engineering artifacts.
- **These lessons can inform policy now.** Policymakers can use them to set clearer expectations for external review, and AI developers can use them to make reviews more credible and useful.

Why perform external review of safety cases?

As frontier AI systems become more capable, there is growing interest in using safety cases to support deployment decisions. A safety case is a structured argument, backed by evidence, that a system is acceptably safe for a particular use (Buhl et al., 2024; Irving, 2024).

Safety cases can help AI developers explain what they believe, why they believe it, and what evidence they are relying on. They can also help regulators, internal decision-makers, and outside reviewers understand the basis for a deployment decision. Safety cases and their review should therefore be seen as instruments for developing understanding. Any policy on their use should emphasise this and not just focus on document production.

But a safety case written by the developer is not enough on its own. If a system could plausibly cause severe harm, the developer has two problems that external review can help with:

1. **Confirmation bias.** Teams can become too attached to their own pre-existing assumptions, methods, and conclusions and may search for, and interpret information in ways that confirm these prior beliefs.
2. **Conflicted incentives.** Developers often face commercial, competitive, and organisational pressure to move forward with deployment.

That does not mean developers act in bad faith. It means that external challenge is necessary if we want safety cases to be decision-relevant rather than just reassuring documents. In other safety-critical sectors, independent scrutiny is normal. AI should not be an exception.

To test the utility of external review in practice, we reviewed Google DeepMind's public **scheming inability safety case** (Phuong et al., 2025). This is a useful example because it is one of the clearest public frontier AI safety-case artifacts available today. DeepMind made a structured argument, used a recognisable Claims-Arguments-Evidence format, published evaluations, and openly discussed some of its own possible weaknesses. That is already better than most public AI safety discussions.

What was good about the Google DeepMind case?

There is real value in the GDM paper. It:

- makes a clear top-level safety argument;
- uses a safety-case structure;
- is unusually transparent about its reasoning;
- and explicitly identifies some weaknesses and limitations.

The paper is also valuable because it gives external reviewers something concrete to inspect. A weak but inspectable safety case is better than vague safety claims that cannot be reviewed at all.

What problems did external review surface?

Our review also found important gaps.

- **The top-level claim was broader than the argument really supported.** The headline claim suggested the model could not cause severe harm even if misaligned, but the supporting argument was really about a narrower class of harm involving scheming in an internal deployment setting.

- **Key terms were not fully defined.** The paper relied heavily on terms like “severe harm,” “deployment Y,” “basic situational awareness,” and “top human-level stealth,” but did not define them precisely enough for external review.
- **The system being assured was not modelled clearly enough.** In practice, the safety-relevant object is not just the base model. It is the wider socio-technical system: prompts, context assembly, tools, memory, permissions, monitors, human oversight, and deployment and operational workflow.
- **The argument for the theory underlying the case was weaker than it first appeared.** The paper relied on the claim that severe harm would require both situational awareness and stealth. We found plausible pathways to harm that did not clearly require both.
- **The evidence was informative, but not strong enough for a broad inability claim.** The evaluations showed something useful about model behavior on particular tasks. But they did not justify the much stronger conclusion that the deployed system lacked the capabilities needed for severe harm. Upper-bounding is challenging and the paper did not justify the elicitation and transfer assumptions strongly enough.
- **Important doubts (known as defeaters in Assurance 2.0) remained unresolved.** These included concerns about oversight, sandbagging, limited evaluation coverage, and the possibility that the deployment environment itself supplies capabilities that the model evaluations assume must be inside the model.

How to review an AI safety case?

We used the **Assurance 2.0** (Bloomfield and Rushby, 2021) framework to structure our review We asked:

- What exactly is being claimed?
- What decision is the claim meant to support?
- What argument connects the evidence to the claim?
- What assumptions does the argument depend on?
- What could make the argument fail?
- How strongly does the evidence really support the conclusion?

From that experience, we think a good external review should include the following steps.

1. Build a common foundation

Reviewers need a shared understanding of the safety-case method being used. For a new team, this may require some initial training on CAE, Assurance 2.0, and the relevant policy context.

2. Start with independent first-pass review

Each reviewer should first examine the safety case independently. This helps reduce groupthink and surfaces different initial doubts.

3. Clarify purpose, scope, and decision relevance

Before examining the evidence, reviewers should ask:

- What decision is this safety case meant to support?
- Is it a real deployment case, a methodological sketch, or a research exercise?
- What part of the risk universe does it cover?
- How does it fit with other safety cases or governance mechanisms?

4. Model the system and the deployment environment

Reviewers need a representation of the actual system being assured. For frontier AI, that usually means more than the AI itself. It includes scaffolding, tools, memory, permissions, monitors, human review, and the operational setting.

5. Analyse risk pathways to harm

A strong review should ask not only whether the safety case is well written, but also how the system could actually cause harm.

This helps in three ways:

- it tests whether the developer's threat model is complete;
- it helps calibrate how much rigour is needed;
- and it identifies which parts of the socio-technical system are safety-relevant.

6. Check the argument structure and logical validity

If the safety case uses a structured method like CAE, reviewers should check whether it is actually being used properly.

That includes checking for:

- unclear or over-broad claims;
- missing warrants and side-claims;
- unsupported assumptions;
- unresolved defeaters;
- and places where there isn't a clear argument for why evidence supports a claim

7. Assess both theory and evidence

The review should examine:

- whether the underlying theory is well supported;
- whether there are counterexamples;
- whether the evaluations really measure the capability they are said to measure;
- and whether results transfer from benchmark tasks to real deployment conditions.

8. Build counter-cases

In our case, three of the most useful counter-cases made the claim that:

- the use of evaluations was not justified as sufficient evidence;
- the reliance on human oversight was not justified;
- and product assurance alone was not enough without process and organisational assurance.

What access to resources do reviewers need?

At a minimum, AI developers should provide:

- the structured safety case itself;
- the decision it is meant to support;
- the exact model or system version under review;
- the deployment context and assumptions;
- a clear definition of the safety property being claimed;
- the expected duration of validity of the case.

Where the case depends on evaluations, developers should also provide:

- details of the prompts, scaffolding, tools, and monitors used;
- thresholds and how they were chosen;
- human baseline methods (if relevant);
- anti-sandbagging or elicitation methods, if any;
- model and system used in the evaluation;
- and the argument for why the evaluation results are supposed to generalize.

Reviewers should also be given access to supporting safety engineering artifacts where they exist, including:

- system models (abstracted representations of the assured system);
- hazard and risk analyses;

- risk pathway analyses;
- safety requirements;
- and an explanation of how this safety case fits with others to cover the wider risk universe.

Where possible, trusted reviewers should also have some form of controlled access to the system or to representative logs and outputs. That will not always be possible publicly, but it is often necessary for serious scrutiny.

Open review vs controlled disclosure

We think there is real value in publishing as much as possible. Publishing argument structure, assumptions, thresholds, and at least part of the evidence base help improve standards across the field.

At the same time, some material may need restricted handling for security or commercial reasons.

The answer is not to hide this silently. It is to publish what can be published, explain what has been withheld, and provide controlled access to trusted reviewers where needed.

Conclusion

External review of frontier AI safety cases is valuable, and it materially improves understanding of risk. Critiques help distinguish between what might be true in principle and what can be confidently established based on evidence.

We found Google DeepMind's scheming safety case to be a thoughtful and valuable first-generation public artefact. However our review found important weaknesses in scope, theory, evidence, and deployment relevance.

If safety cases are to support real decisions, and if developers want their cases to be credible, then external review should be treated as a normal part of frontier AI assurance, not an optional extra. To that end, our work provides actionable recommendations to policymakers on how effective external review can be undertaken.