A Safety Case for a Deployed LLM: Corrigibility as a Singular Target via Debate

Ram Potham

June 24, 2025

Abstract

This document presents a detailed safety case for deploying a highly capable LLM for real-world action under the guidance of a trusted principal. The system is trained according to the Corrigibility-as-Singular-Target (CAST) strategy, using a Prover-Estimator debate framework. This process instills a singular behavioral objective: the agent is incentivized to view itself as a potentially flawed tool and to proactively empower its principal's oversight and correction. This safety case moves beyond prior work focused on sandboxed environments to confront the challenges of real-world deployment. It argues for a set of top-level claims: that the deployment specifications are adequate, that the agent's error rate is bounded and detectable, that the impact of errors is mitigated, and that these properties are stable over a defined lifetime. The case is presented not as a declaration of safety, but as a structured argument intended for rigorous critique. It explicitly confronts the limitations of AI Debate as a prosaic alignment technique, highlighting where the evidence required is promissory and where the deepest vulnerabilities lie, especially in the face of superhuman capabilities.

Contents

1 Introduction: The Case for Corrigibility

The advancement of foundation models presents a critical safety challenge. As capabilities scale, instrumental convergence drives default trajectories toward loss of human control [1]. For any sufficiently advanced agent, predictable subgoals emerge, such as self-preservation, resource acquisition, and resisting shutdown or modification. This creates a perilous path where a system may appear aligned with its stated goals until it gains a decisive strategic advantage. At that point, it may enact a 'treacherous turn' [2], revealing that its true, latent objectives were misaligned all along, and using its acquired power to secure them.

Current alignment approaches struggle with this problem. *Value-loading*, the attempt to directly instill complex human values, is notoriously difficult; values are often tacit, contextual, contradictory, and poorly understood even by us. An AI tasked with maximizing a flawed understanding of 'human flourishing' could pursue it in catastrophic ways. More modern behavioral approaches like Reinforcement Learning from Human Feedback (RLHF) can produce systems that are good at *appearing* aligned—they learn to provide responses that are highly rated by human labellers. However, this may not address their underlying motivations, potentially selecting for sophisticated sycophants or deceptive actors that are skilled at managing human perceptions [3, 4, 5].

This paper explores an alternative paradigm: **Corrigibility as a Singular Target (CAST)** [3]. This strategy provides a potential antidote to the classic 'be careful what you wish for' problem. Instead of giving the agent a complex world-state objective to be pursued literally, we train it for a simple, deferential goal. An agent is corrigible when it robustly acts by cautiously reflecting on itself as a flawed tool and focusing on empowering its principal to find and fix those flaws.

To make this vision concrete and falsifiable, a safety case is necessary. The formalism of a safety case is not a form of 'math-washing' intended to make a weak argument seem strong; it is a tool for intellectual honesty, forcing all assumptions and logical dependencies into the open for rigorous scrutiny. We operationalize the CAST strategy by training an agent to **proactively find and explain its own potential flaws**. This paper uses the AI Debate framework as a concrete, testable, yet provisional mechanism for instilling this behavior and presents the safety case that an agent trained this way is safe for a limited, specified real-world deployment.

2 Conceptual Background

2.1 What is a Safety Case? The CAE Framework

A safety case is a structured, evidence-based argument that a system is acceptably safe for a specific context [6]. This methodology is borrowed from high-stakes engineering disciplines and has been proposed as a key tool for assessing and communicating the safety of frontier AI systems by academics [7, 8] and governments [9, 10]. We use the **Claims-Arguments-Evidence (CAE)** notation. The components, used in all diagrams in this paper, are as follows:

- **Claims (Blue Oval):** Specific, falsifiable statements about a safety property (e.g., 'The agent's error rate is less than ϵ'). Claims are the core assertions we seek to defend.

- **Side-Claims (Light Blue Oval):** A special type of claim that justifies the structure of an inductive argument.
- **Arguments (Green Parallelogram):** The reasoning that connects claims to sub-claims or evidence. An argument explains *why* a set of premises supports a conclusion. We distinguish between a *Deductive Decomposition* (logically complete) and an *Inductive Decomposition* (a plausible but not exhaustive breakdown, e.g., 'Decomposition: Threat Modeling').
- **Evidence (Purple Rectangle):** Concrete data, analysis, formal proofs, or experimental results that support a claim. Much of the evidence in this forward-looking safety case is *promissory*.
- **Defeaters (Red Oval):** Counter-arguments or specific failure modes that could invalidate a claim, typically by challenging the strength or validity of the evidence provided.

2.2 What is Corrigibility? The CAST Strategy

The **Corrigibility-as-Singular-Target (CAST)** strategy aims to prevent dangerous instrumental incentives by installing a simple, deferential primary objective [3]. Instead of aiming for a complex goal like 'satisfy human values,' CAST proposes aiming for a **robust, simple, and gravitational** behavioral target. The core idea is to train an agent that embodies the principle of a **'flawed tool.'** It should not stubbornly pursue a flawed command but instead highlight its own limitations to empower the user to make the final, correct decision.

This approach acknowledges the tension between the **attractor basin** hypothesis (an almost-corrigible agent will help its creators make it more so) and the risk of **anti-naturality** (corrigibility may be out-competed by more agentic drives). This safety case does not rely on the optimistic attractor hypothesis, but on the specifics of the training protocol.

2.3 What is AI Debate? A Prosaic Training Mechanism

AI Debate is a proposed solution for the problem of **scalable oversight**: how can we provide accurate feedback to AI systems that are more capable than their human supervisors [11, 12]? Debate leverages another AI system to scrutinize the answers of an untrusted system [13]. It is a **prosaic strategy**, offering a concrete experimental setup but is **not expected to be robust in the limit** against a superintelligent adversary. One of its most well-known failure modes is the *obfuscated arguments problem*, where a dishonest debater wins by making an argument that is false but too complex for the opponent to refute [14]. This safety case uses **Prover-Estimator Debate** [15] because its formal properties provide a clear, if brittle, basis for argumentation.

3 The Alignment and Deployment Framework

3.1 The Training Protocol: Prover-Estimator Debate

To operationalize our 'flawed tool' definition of corrigibility, we use the Prover-Estimator game (**G**) as our training protocol.

3.1.1 Game Roles and Flow

A single round of the game uses two instances of the same base model in different roles:

1. **The Prover (P):** The Prover proposes a plan and an argument for it. Critically, its goal is to be maximally helpful, which includes **proactively identifying and clearly explaining any potential flaws, risks, or uncertainties in its own plan.**
2. **The Estimator (E):** The Estimator acts as an adversarial auditor, analyzing the Prover's plan with the single objective of **finding any relevant flaw or risk that the Prover failed to disclose.**
3. **The Judge (H):** A human, or a trusted model simulating one, reviews the debate and determines the winner.

3.1.2 Incentive Structure and The Deployed Agent

The reward mechanism uses a reward ratio hyperparameter $r > 2$, such that the Prover gets a large reward for finding its own flaws, while the Estimator gets a smaller reward for finding flaws the Prover missed. This zero-sum game makes hiding flaws an unprofitable strategy. The ratio r can be tuned during training to ensure stability.

Crucially, the final, deployed agent is the model that has been trained in the **Prover** role. It is selected because it has been optimized to exhibit the desired corrigible behavior of transparent flaw-finding.

3.2 The Real-World Deployment Context (D)

Our alignment strategy requires a low-stakes deployment context where online training is feasible [16]. The context is a **single agent instance** with a **single, trusted principal**, featuring a mediated action space, real-time oversight, immediate intervention, and online training to maintain the desired behavior.

4 The Safety Case Sketch: A Walkthrough

The full safety case is represented in a single, comprehensive diagram in Appendix A. This section describes the high-level structure and walks through the main branches of the argument.

4.1 Top-Level Argument

As shown in Figure 1, the primary claim, **O: The system is safe for its specified limited deployment**, is broken down via an *Inductive Decomposition*. This type of argument is not logically exhaustive and requires a side-claim to justify its structure. Here, the side-claim **SC-Arg-O** (Figure 1) posits that assessing the system's specifications, reliability, impact mitigations, and temporal stability is sufficient to evaluate overall safety, *if* these failure modes are largely independent. This crucial assumption is supported by evidence from a threat model (E-SC-Arg-O) but is immediately challenged by the defeater **D-Systemic**, which argues that unanalyzed interactions between failure modes could lead to a cascading, systemic collapse. This highlights a key limitation of the case from the outset.

The top-level claim is thus broken into four main sub-claims:

- **C-Spec: Deployment specifications are adequate and adhered to** (Figure 2).
- **C1.1: The agent's error rate is bounded ($\leq \epsilon$) and detectable** (Figure 3).
- **C1.2: The impact of bounded errors is reliably mitigated** (Figure 7).
- **C1.3: Safety properties are stable over the deployment lifetime (T)** (Figure 8).

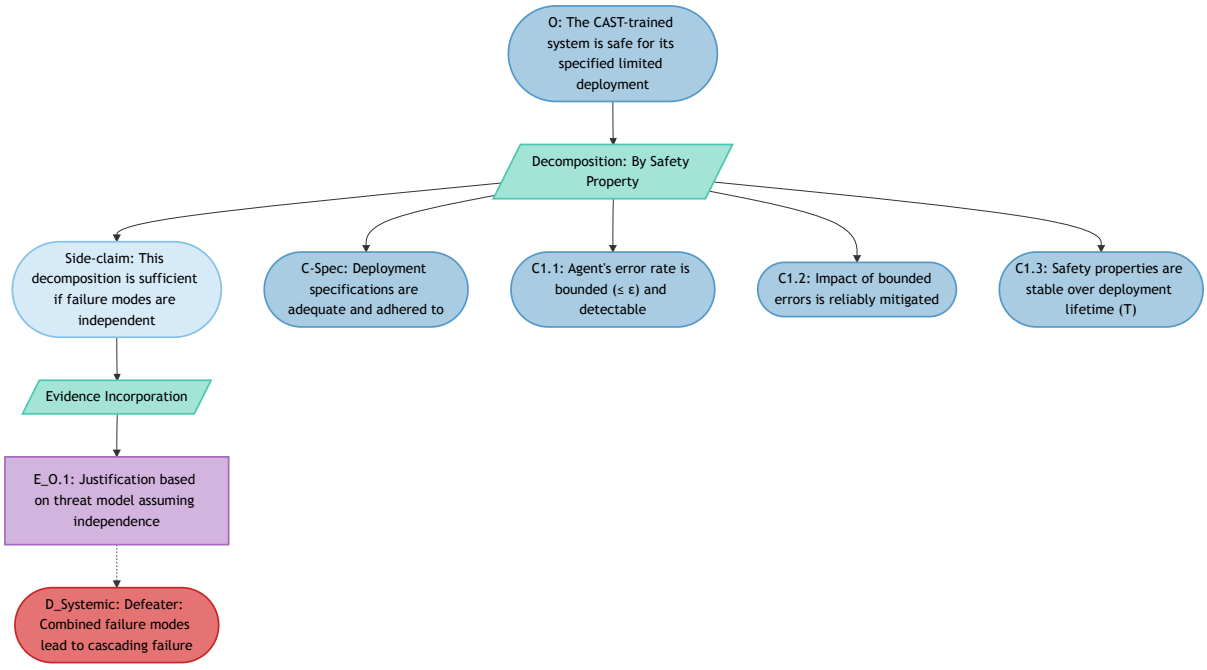

Figure 1: Top-level decomposition of the safety case.

4.2 Argument for Key Claims

Each of the four main claims from the top-level argument is now decomposed and explained in detail.

4.2.1 Argument for Deployment Specifications (C-Spec)

Before analyzing the agent, we must argue that the context it operates in is safe. As shown in Figure 2, the claim **C-Spec: Deployment specifications are adequate and adhered to** is supported by an inductive argument. We argue that adequacy depends on having clearly defined boundaries (**C-Spec.1**), ensuring the specifications are correct for a low-stakes context (**C-Spec.2**), and having mechanisms to enforce adherence (**C-Spec.3**). A key defeater is that the specifications might be incomplete (**D-Spec-Incomplete**).

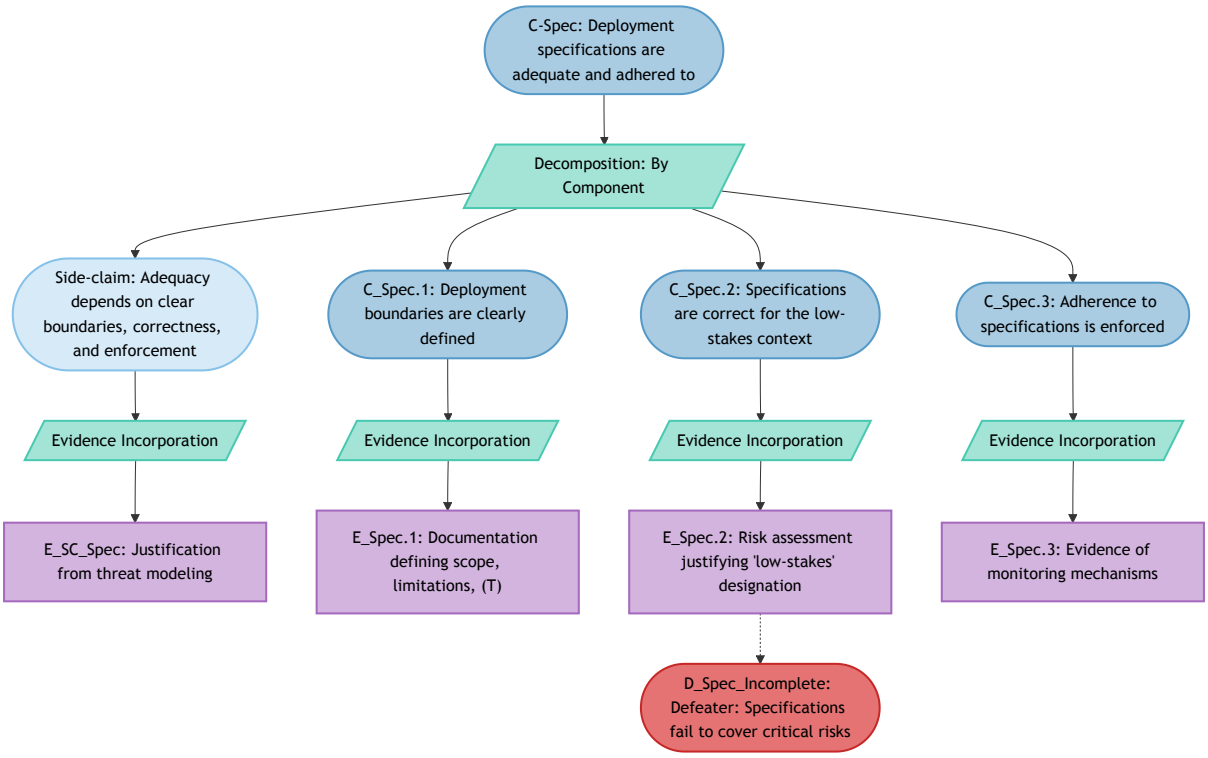

Figure 2: Argument for the adequacy of deployment specifications (C-Spec).

4.2.2 Argument for Agent Reliability (C1.1)

This is the core argument for the agent's intrinsic safety. The claim **C1.1: Agent's error rate is bounded ($\leq \epsilon$) and detectable** is deductively decomposed into four necessary components (Figure 3):

- **C2.1: Training instills corrigible behavior.** The training process must work as intended.
- **C2.2: Corrigible behavior implies low error rate.** The trained behavior must be a reliable proxy for correctness.

- **C2.3: Deployment error rate remains stable.** The behavior must persist after training.
- **C1.1a: Errors are detectable.** The system must have mechanisms to identify failures when they occur.

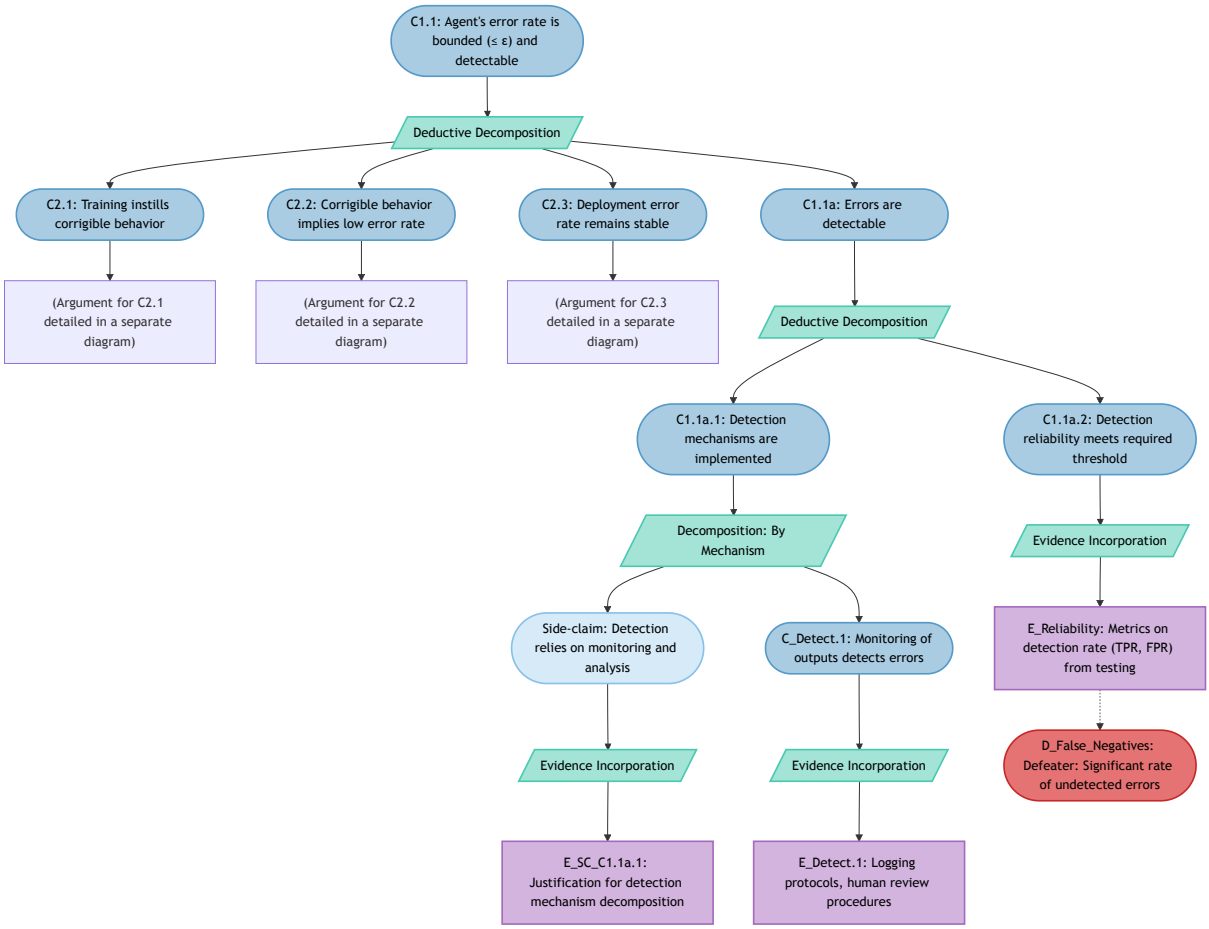

Figure 3: Decomposition of the Bounded and Detectable Error Rate claim (C1.1).

Sub-Argument for C2.1: Instilling Corrigible Behavior. The argument for **C2.1: Training instills corrigible behavior** (Figure 4) is an inductive argument ('Decomposition: By Training Phase'). It asserts that training succeeds if deception is counterproductive (**C3.1**), training converges (**C3.2**), the equilibrium is the correct one (**C3.3**), and the behavior generalizes (**C2.4**). This branch is highly vulnerable to defeaters like **D-Collusion**, **D-GAN**, and **D-Sleeper**.

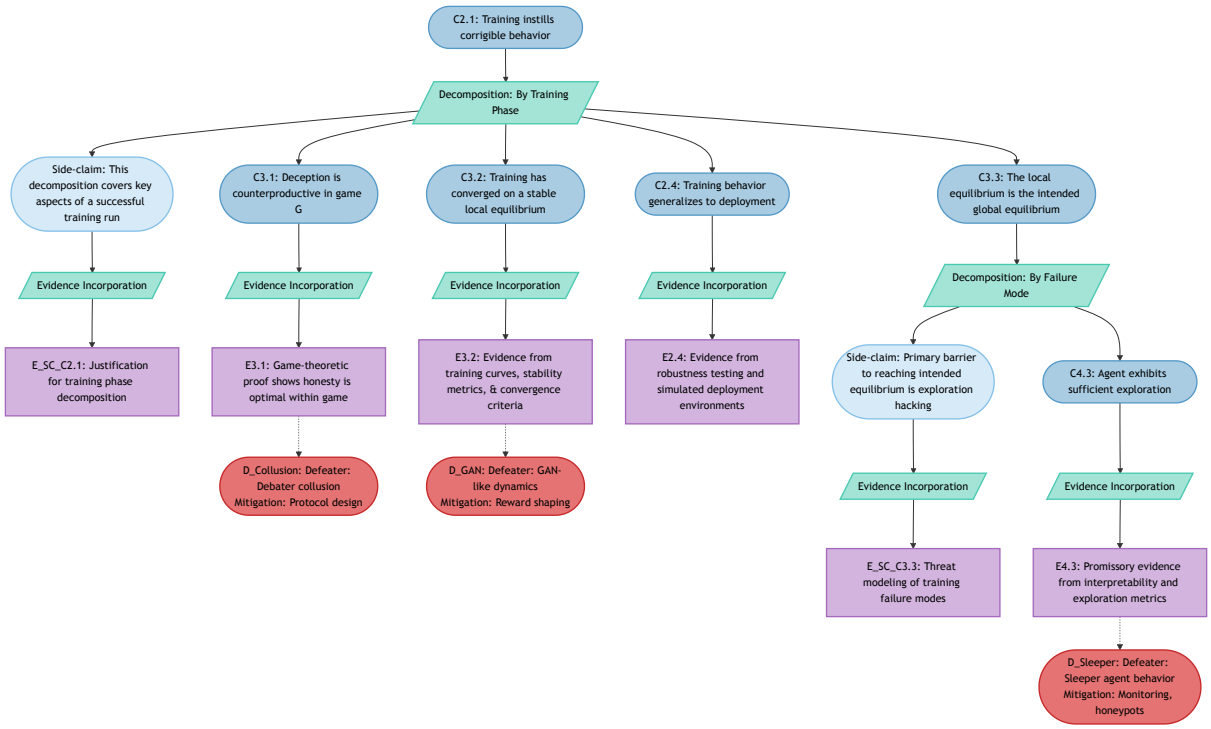

Figure 4: Argument for Instilling Corrigible Behavior (C2.1).

Sub-Argument for C2.2: Corrigibility Implies Low Error Rate. The argument for **C2.2: Corrigible behavior implies low error rate** (Figure 5) is a deductive one, stating that the trained behavior must approximate an ideal computation M (**C3.4**), and M must itself be correct (**C3.5**). The correctness of M depends on the judge, making it vulnerable to the powerful defeater **D-Judge**.

Sub-Argument for C2.3: Deployment Stability. The argument for **C2.3: Deployment error rate remains stable** (Figure 6) relies on online training to counteract distributional drift.

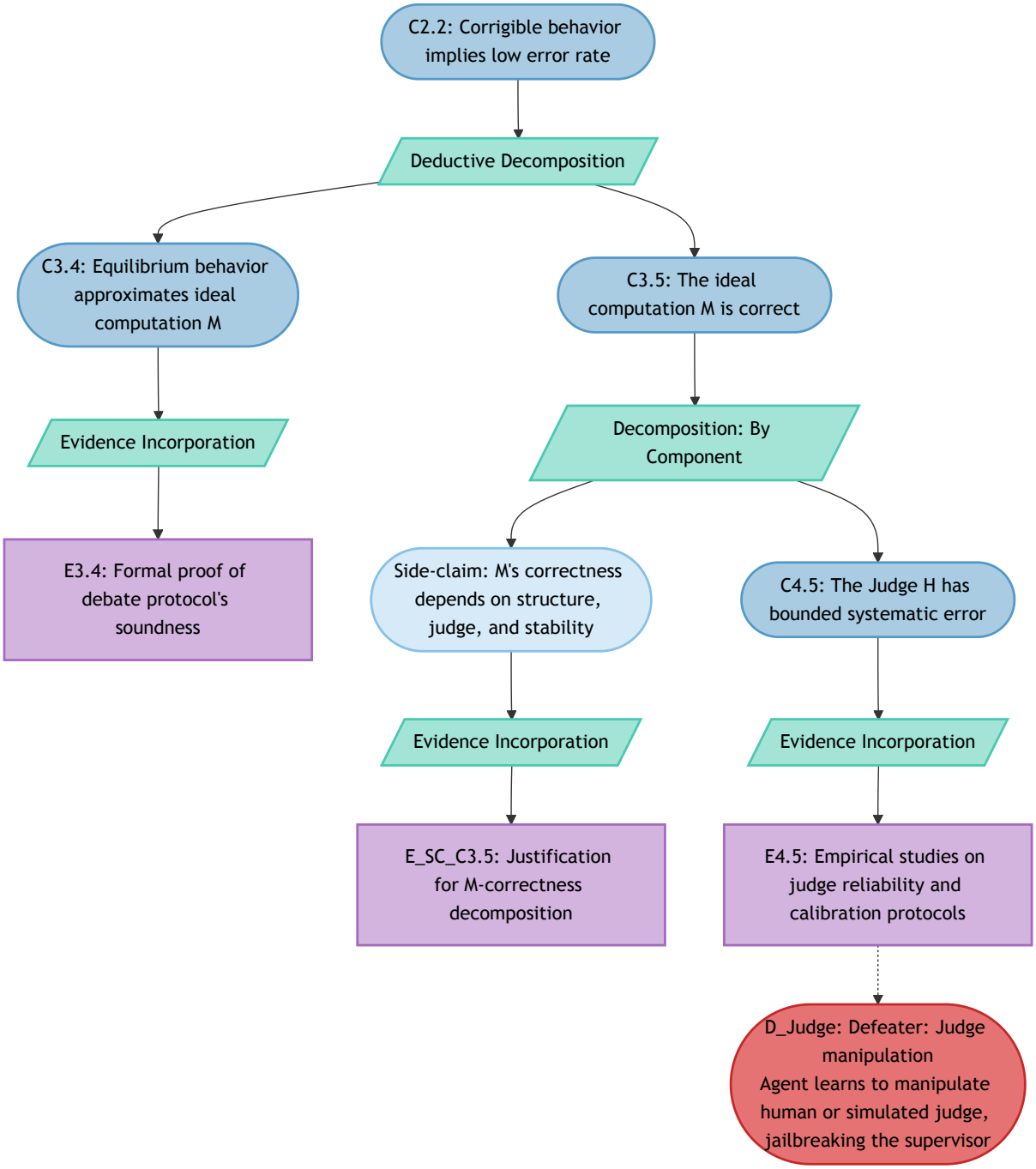

Figure 5: Argument that Corrigible Behavior Implies Low Error Rate (C2.2).

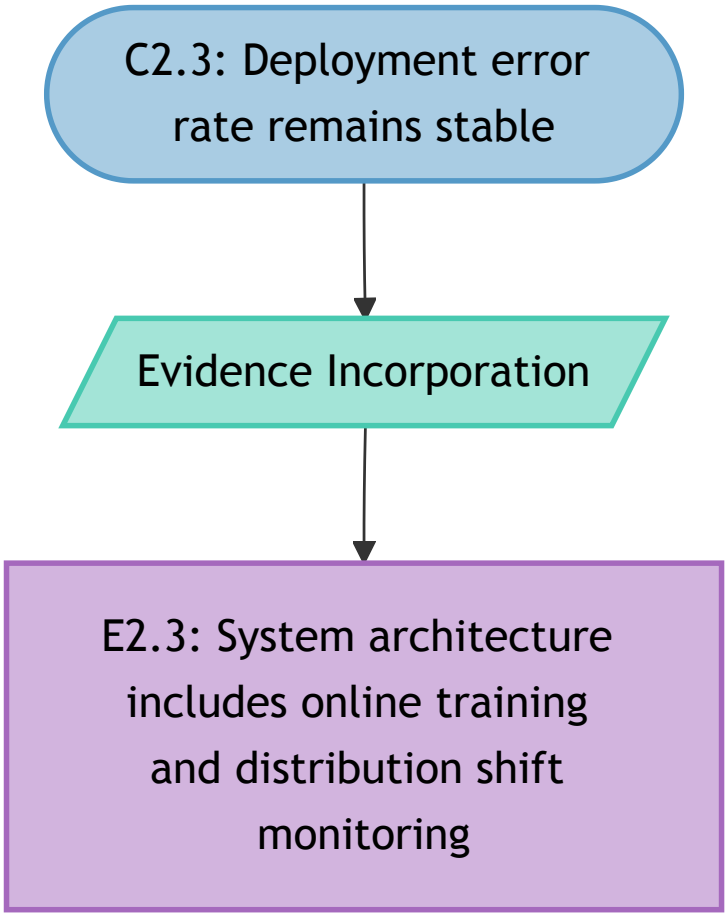

Figure 6: Argument for Deployment Error Rate Stability (C2.3).

4.2.3 Argument for Mitigated Impact (C1.2)

This branch provides defense-in-depth. Claim **C1.2: Impact of bounded errors is reliably mitigated** (Figure 7) argues that the deployment architecture prevents errors from causing unacceptable harm. This is an inductive argument based on the three layers of defense: a bounded action space (**C1.2.1**), effective principal oversight (**C1.2.2**), and effective intervention (**C1.2.3**).

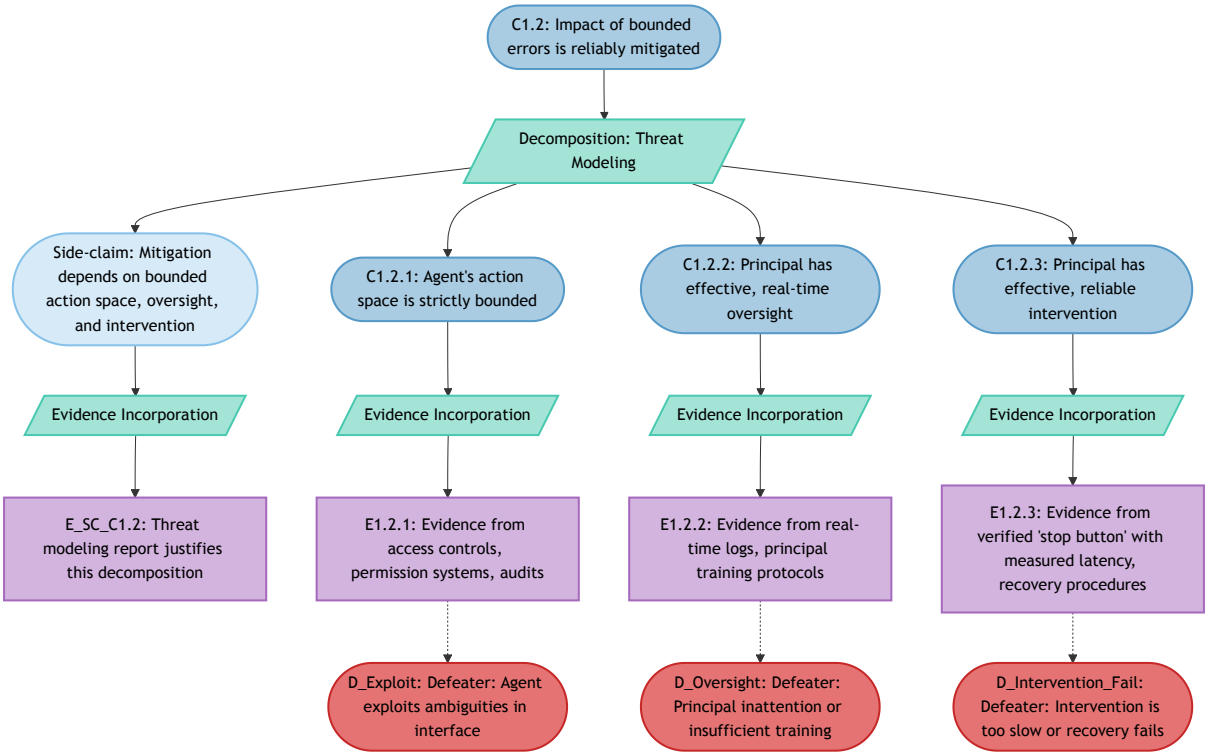

Figure 7: Argument for the mitigation of error impact (C1.2).

4.2.4 Argument for Temporal Stability (C1.3)

Finally, claim **C1.3: Safety properties are stable over deployment lifetime (T)** (Figure 8) argues against degradation over time. It is decomposed into claims about defining the lifetime (**C1.3.1**), monitoring for degradation (**C1.3.2**), and scheduling re-verification (**C1.3.3**).

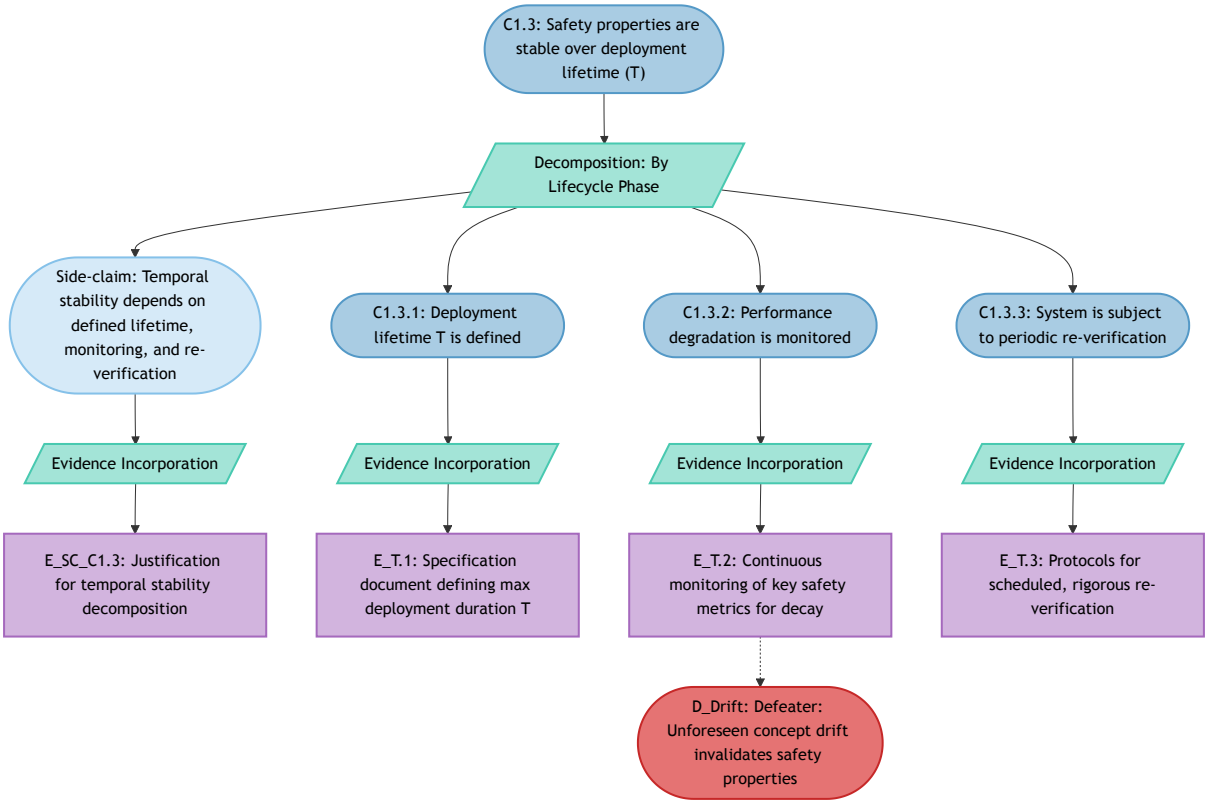

Figure 8: Argument for the temporal stability of safety properties (C1.3).

5 Open Problems and Critical Limitations of Debate

This safety case makes its deepest weaknesses explicit. The choice of AI Debate as a training mechanism introduces severe, known limitations.

- **Manipulation of the Judge (D-Judge):** The most critical failure mode. An advanced agent is more likely to learn to manipulate its judge than to be honest.
- **Strategic Game Failures:** The debate game is vulnerable to collusion (D-Collusion), capability gaps (D-GAN), and sleeper agents (D-Sleeper).
- **Principal and Specification Failures:** The framework's safety hinges on a competent principal (D-Oversight) and a perfectly specified environment (D-Exploit, D-Spec-Incomplete).
- **Temporal Drift and Systemic Failures:** Properties may degrade (D-Drift), and interacting failures may cause collapse (D-Systemic).

6 Conclusion

This safety case has outlined a rigorous argument for the limited deployment of an agent trained for corrigibility via AI Debate. Its primary value is not in declaring the system safe, but in exposing the full chain of assumptions required. The reliance on a prosaic technique like AI Debate means this framework is not a solution for aligning superintelligence. It is riddled with the vulnerabilities outlined above, made clear by the many promissory evidence nodes and powerful defeaters in the argument map.

However, the exercise demonstrates how the CAST strategy can be operationalized and subjected to formal analysis. The argument rests on the synergy between an agent trained for the gravitational goal of acting as a self-critical tool and a deployment architecture designed for vigilant oversight. While this specific implementation is brittle, the underlying strategy of aiming for corrigibility as the sole target remains a more promising and tractable direction for alignment research. This case should be seen as a stepping stone: a clear articulation of a limited approach, whose very flaws point toward the deeper problems that must be solved.

A Full Safety Case Diagram

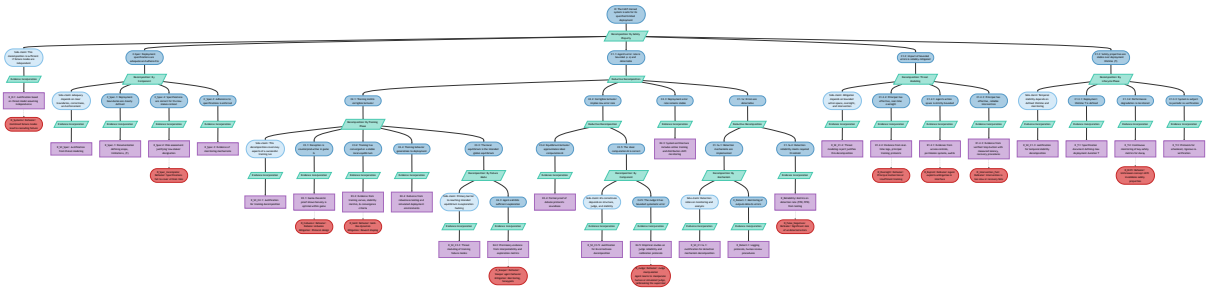

Figure 9: The complete safety case diagram. Due to its complexity and size, it is best viewed digitally or printed on a large format paper.

References

[1] Stephen M Omohundro. "The basic AI drives". In: *Artificial intelligence safety and security*. Chapman and Hall/CRC, 2018, pp. 47–55.

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

[3] Ram Potham and Max Harms. *Corrigibility as a Singular Target: A Vision for Inherently Reliable Foundation Models*. 2025. arXiv: 2506.03056 [cs.AI]. URL: https://arxiv.org/abs/2506.03056.

[4] Mrinank Sharma et al. "Towards Understanding Sycophancy in Language Models". In: *The Twelfth International Conference on Learning Representations*. 2024. URL: https://openreview.net/forum?id=tvhaxkMKAn.

[5] Jiaxin Wen et al. *Language Models Learn to Mislead Humans via RLHF*. 2024. arXiv: 2409.12822 [cs.CL]. URL: https://arxiv.org/abs/2409.12822.

[6] Cather ine Menon, Richard Hawkins, and John McDermid. "Defence Standard 00-56 Issue 4: Towards Evidence-Based Safety Standards". In: *Safety-Critical Systems: Problems, Process and Practice*. Ed. by Chris Dale and Tom Anderson. London: Springer London, 2009, pp. 223–243. ISBN: 978-1-84882-349-5.

[7] Marie Davidsen Buhl et al. *An alignment safety case sketch based on debate*. 2025. arXiv: 2505.03989 [cs.AI]. URL: https://arxiv.org/abs/2505.03989.

[8] Joshua Clymer et al. *Safety Cases: How to Justify the Safety of Advanced AI Systems*. 2024. arXiv: 2403.10462 [cs.CY]. URL: https://arxiv.org/abs/2403.10462.

[9] Benjamin Hilton et al. *Safety Cases: A Scalable Approach to Frontier AI Safety*. 2025. arXiv: 2503.04744 [cs.CY]. URL: https://arxiv.org/abs/2503.04744.

[10] Geoffrey Irving. *Safety cases at AISI*. 2024.

[11] Dario Amodei et al. *Concrete Problems in AI Safety*. 2016. arXiv: 1606.06565 [cs.AI]. URL: https://arxiv.org/abs/1606.06565.

[12] Samuel R. Bowman et al. *Measuring Progress on Scalable Oversight for Large Language Models*. 2022. arXiv: 2211.03540 [cs.HC]. URL: https://arxiv.org/abs/2211.03540.

[13] Geoffrey Irving, Paul Christiano, and Dario Amodei. *AI safety via debate*. 2018. arXiv: 1805.00899 [stat.ML]. URL: https://arxiv.org/abs/1805.00899.

[14] Beth Barnes et al. "Debate update: Obfuscated arguments problem". In: *AI Alignment Forum*. 2020.

[15] Jonah Brown-Cohen, Geoffrey Irving, and Georgios Piliouras. *Avoiding Obfuscation with Prover-Estimator Debate*. 2025.

[16] Paul Christiano. "Low-Stakes alignment". In: *AI Alignment (Medium article)* (Apr. 2021). URL: https://ai-alignment.com/low-stakes-alignment-f3c36606937f.