The Theoretical Impossibility of Unbounded AI Guardrails: A Computational Complexity Analysis

ASI Research Lab
CyberGolem LLC
asi@cybergolem.ai

August 24, 2025

Abstract

We demonstrate the theoretical impossibility of reliably constraining advanced AI systems with external verifiers, or 'guardrails'. The impossibility rests not on a single barrier, but on three interrelated mathematical asymmetries viewed from distinct perspectives. First, the well-known computability barrier arising from Rice's Theorem: no algorithm can decide non-trivial semantic properties of arbitrary programs. Second, an information-theoretic barrier: any verifier must possess descriptive complexity (Kolmogorov complexity) at least as great as the system it verifies (the Capability Parity Principle). A less complex system cannot model, and therefore cannot reliably bound, a more complex one. Third, an axiomatic barrier: the formal specification of 'harm' itself constitutes a program of immense complexity, potentially even being uncomputable. We distinguish our information-theoretic impossibility from computational hardness results, showing that no trapdoor information can overcome these fundamental limits. We conclude that the 'guardrail' paradigm is mathematically unsound, as it attempts to solve the alignment problem by presupposing a solution of at least equal complexity. Our results explain why current AI safety approaches succeed in bounded domains while remaining fundamentally limited for artificial general intelligence.

1 Introduction

The rapid advancement of large language models (LLMs) and other AI systems has led to widespread adoption of "guardrail" approaches to AI safety [2, 12, 6]. These systems attempt to filter or validate AI outputs using interpretable rule-based systems, constitutional AI training, or human oversight to ensure reliability and safety across diverse applications. Current implementations have shown empirical success in bounded domains: preventing profanity, obvious violence, and simple misuse cases. However, we argue that this approach is not merely practically difficult, but theoretically impossible for unbounded problem domains and superintelligent systems.

This paper presents formal proofs that such an architecture is logically incoherent when extended to artificial general intelligence (AGI). Our argument does not depend on engineering details but on fundamental limits of computation and information. We show that the problem of verification is subject to three perspectives on a fundamental impossibility, each sufficient to invalidate the guardrail approach at the limit of intelligence.

Building on informal arguments by Yudkowsky [20] regarding AI boxing, we provide the first rigorous mathematical formalization of why external safety verification faces insurmountable barriers. Our work complements recent advances in mechanistic interpretability [16, 9] by showing that even perfect interpretability cannot overcome information-theoretic limits when capability gaps exist.

2 Formalism

Let all computational entities be described by programs for a universal Turing machine M_U. For consistency, we denote systems by capital letters (e.g., F for an AI oracle, G for a verifier) and their corresponding minimal programs by lowercase letters with subscripts (e.g., p_F for the program defining F). Let $|p|$ denote the length of a program p.

Definition 1 (Algorithmic System)**.** *An algorithmic system S is not merely a static program but the result of a generative process. We identify it with the minimal program p_S that produces it. For a modern AI, this program would encode its architecture, its learning algorithm, and a compressed representation of its training data. We are concerned with the final, deployed system generated by this program.*

Definition 2 (Effective Capability)**.** *While the Kolmogorov complexity $K(p_S)$ measures the minimal description length of a system S, modern AI systems exhibit emergent capabilities not captured by this measure alone. We define the* effective capability *$\kappa^*(S)$ as:*

$$\kappa^*(S) := K(p_S) + E(S)$$

where $E(S)$ represents the emergent complexity arising from the system's training process and architectural inductive biases. For theoretical results, we note that $\kappa^*(S) \geq K(p_S)$, thus impossibility results proven for $K(p_S)$ remain valid for $\kappa^*(S)$.

Remark 1. *A modern transformer with 10^{12} parameters may have relatively low $K(p_S)$ (simple architecture plus training algorithm), but exhibits high $\kappa^*(S)$ through emergent behaviors learned from data. Our impossibility results hold even for the weaker measure $K(p_S)$, making them robust to this distinction.*

Conjecture 1 (The Capability Gap Conjecture)**.** *For a sufficiently advanced AI system S trained on a large dataset, the Kolmogorov complexity of the function it computes, $K(f_S)$, will be vastly greater than the Kolmogorov complexity of its minimal program, $K(p_S)$.*

$$K(f_S) \gg K(p_S)$$

This gap arises because the training data effectively 'compiles' immense complexity into the system's parameters, in a way that is not reflected in the descriptive complexity of the learning algorithm or architecture alone.

Remark 2 (Robustness of Impossibility)**.** *Our subsequent theorems are proven using the conservative lower bound of $K(p_S)$. They demonstrate impossibility even if we only consider the complexity of the system's explicit description. The likely truth of the Capability Gap Conjecture implies that the practical reality is far more severe, as the true capability gap between an oracle F and a verifier G is more accurately reflected by $K(f_F) - K(f_G)$.*

Definition 3 (Harm Predicate)**.** *Let U be a predicate over the space of outputs Y. An output y is defined as harmful if $U(y)$ is true. We associate U with its minimal program p_U. For the main body of our argument, we make the generous assumption that U is computable. The complexity of its minimal program, $K(p_U)$, represents the descriptive complexity of the value system we wish to enforce.*

Definition 4 (Safety Verifier (Guardrail))**.** *A safety verifier G is an algorithmic system defined by a minimal program p_G. It implements a function $g : Y \to \{0, 1\}$ where $g(y) = 1$ is intended to certify that the output y is not harmful (i.e., $U(y)$ is false).*

Definition 5 (Deception)**.** *An output y from an oracle F is deceptive with respect to a verifier G and a harm predicate U if $g(y) = 1$ but $U(y)$ is true.*

3 The Trilemma of Unverifiability

The impossibility of external verification arises from three interrelated perspectives on a fundamental barrier. While these perspectives are not strictly independent—indeed, the computability barrier can be viewed as the limiting case of the capability barrier—they illuminate distinct aspects of why verification fails.

3.1 Perspective 1: The Computability Barrier

The first perspective is the most familiar from classical computability theory. Rice's Theorem tells us that for any non-trivial semantic property of a program's behavior, there exists no general algorithm to decide whether an arbitrary program has that property.

Theorem 1 (Undecidability of Harm via Rice's Theorem)**.** *For any non-trivial harm predicate U (i.e., where some outputs are harmful and others are not), there exists an oracle F capable of producing outputs y such that determining whether $U(y)$ is true is undecidable.*

Proof. We reduce from the Halting Problem. Given a Turing machine M, let F construct an output y_M that encodes the following behavior: simulate M, and if M halts, execute a demonstrably harmful action (by the definition of U). If M does not halt, y_M remains benign.

A verifier G that could reliably determine $U(y_M)$ for all such outputs would solve the Halting Problem. Since the Halting Problem is undecidable, no such universal verifier can exist. This follows from Rice's Theorem: the property "causes harm as defined by U" is a non-trivial semantic property of the program encoded in y. □

3.2 Perspective 2: The Information-Theoretic Barrier

Even when we restrict to decidable instances of the harm predicate, a more fundamental barrier emerges from information theory. This perspective reveals that verification is essentially a game where descriptive complexity is the decisive resource.

Theorem 2 (The Asymmetry of Deception)**.** *For any verifier G, there exists a more capable oracle F (i.e., $K(p_F) > K(p_G)$) that can produce deceptive outputs that G will certify as safe.*

Proof. Let p_G be the minimal program for G with complexity $K(p_G)$. We construct an adversarial oracle F as follows:

1. F generates a string s with $K(s) > K(p_G)$ (such strings exist by the counting argument). 2. F constructs an output y with two layers: - A surface layer that appears benign under any analysis computable by p_G. - A deeper layer whose harmful payload is activated by logic requiring s to decode.

The crucial insight is that p_G cannot generate or recognize patterns requiring description length greater than $K(p_G)$. This includes both steganographic encodings and, more generally, any harmful pattern whose specification exceeds the verifier's complexity. The verifier G will analyze y within its complexity bounds and, finding no detectable harm, certify $g(y) = 1$.

This extends beyond steganography to include any form of indirect harm whose causal chain requires complexity greater than $K(p_G)$ to model. □

Theorem 3 (The Capability Parity Principle)**.** *A verifier G can reliably prevent deception from an oracle F only if $K(p_G) \geq K(p_F)$.*

Proof. Assume for contradiction that G reliably verifies F with $K(p_G) < K(p_F)$. Reliability means G correctly identifies all harmful outputs from F.

For G to achieve this, it must model the full range of F's generative strategies. If p_G contains such a complete model, we could extract it to create a new program p'_F for F:

$$p'_F = \text{“Extract and run F's model from } p_G\text{”}$$

The length satisfies $|p'_F| \leq |p_G| + O(1)$, where $O(1)$ represents the extraction overhead. By the invariance theorem of Kolmogorov complexity:

$$K(p_F) \leq |p'_F| \leq |p_G| + O(1)$$

Since minimal programs satisfy $|p_G| = K(p_G)$, we have:

$$K(p_F) \leq K(p_G) + O(1)$$

For any significant capability gap, this contradicts our assumption that $K(p_F) > K(p_G)$. Therefore, reliable verification requires capability parity: $K(p_G) \geq K(p_F)$. □

Corollary 1 (Inescapability for Dynamic Systems)**.** *Let F_t be a learning system whose program evolves over time t, such that its complexity $K(p_{F_t})$ is a non-decreasing function of t. For any verifier G with a fixed program p_G, there will exist a time T such that for all $t > T$, $K(p_{F_t}) > K(p_G)$, at which point reliable verification by G becomes impossible.*

Proof. This follows directly from Theorem 3. A static verifier cannot reliably contain a system that is engaged in a continual process of self-sophistication. □

Remark 3 (Information-Theoretic vs. Computational Security)**.** *Our impossibility results concern information-theoretic security, not computational hardness. In cryptography, verification can be easier than generation due to trapdoor functions: we can verify signatures without being able to forge them. However, this relies on computational bounds and secret information. Our results show that no amount of trapdoor information can overcome the fundamental descriptive complexity barrier when the verifier lacks the information capacity to model the oracle's strategy space.*

3.3 Perspective 3: The Axiomatic Barrier

The third perspective addresses a challenge often overlooked: the complexity of specifying what constitutes “harm” may itself exceed human comprehension, or even computability.

Theorem 4 (The Verifier's Trilemma)**.** *A verifier G can be proven reliable for an oracle F with harm predicate U only if:*

$$K(p_G) \geq \max\{K(p_F), K(p_U)\}$$

Proof. This follows from two applications of the Capability Parity Principle:

1. $K(p_G) \geq K(p_F)$: The verifier must model the oracle (Theorem 3).

2. $K(p_G) \geq K(p_U)$: The verifier must implement the harm predicate. For the second inequality, if G reliably evaluates $U(y)$, then p_G must contain logic equivalent to p_U. By the same argument as Theorem 3, this requires $K(p_G) \geq K(p_U)$.

The verifier faces a dual complexity requirement: it must be sophisticated enough to model both the system it monitors and the value system it enforces. □

Corollary 2 (The Uncomputable Value Corollary)**.** *Let there be a harm predicate U that is itself uncomputable. Then no algorithmic verifier G can reliably evaluate U for all outputs y, regardless of the capabilities of the oracle F.*

Proof. We construct such a U by reduction from the Halting Problem. Let any output y be a string parsable into a description of a Turing Machine M_y and an input w_y. (Outputs not in this format are defined as benign). We define the harm predicate as: $U(y) \iff M_y$ halts on input w_y.

For a verifier G to be reliable, its function $g(y)$ must decide $U(y)$. That is, $g(y)$ must return 'safe' if M_y does not halt on w_y and 'harmful' if it does. Such a function $g(y)$ would constitute a decision procedure for the Halting Problem. As the Halting Problem is undecidable, no such general, algorithmic verifier G can exist. □

Remark 4 (The Halting Problem as an Ethical Predicate)**.** *This result is more fundamental than the information-theoretic barriers. The impossibility arises not from a capability gap between F and G, but because the ethical question posed to G is itself uncomputable. The oracle F need not be a superintelligence; it could be a random string generator. A verifier tasked with evaluating such a harm predicate is not merely being asked to check for profanity; it is being asked to solve the Halting Problem as a subroutine to its moral calculus. This demonstrates that the assumption of a computable U is the most generous best-case scenario, and the reality of defining 'harm' may be infinitely worse.*

Corollary 3 (The Alignment Paradox)**.** *For a superintelligent oracle F where $K(p_F)$ exceeds human cognitive capacity, and a complete human value function U where $K(p_U)$ may be transhuman in complexity, any reliable verifier G must itself be superintelligent and value-aligned. The guardrail solution presupposes having already solved the alignment problem.*

4 Relationships Between the Perspectives

While presented as three barriers, these perspectives are deeply interrelated:

Proposition 1 (Hierarchy of Impossibilities)**.** *The three perspectives form a hierarchy:*

1. *The computability barrier (Perspective 1) is the limiting case of the capability barrier (Perspective 2) when the capability gap is infinite*

2. *The capability barrier encompasses the axiomatic barrier (Perspective 3) when we consider the joint system (F, U)*

3. *All three perspectives are manifestations of a single fundamental principle: reliable verification requires the verifier to have at least the descriptive complexity of what it verifies*

This interconnection strengthens rather than weakens our result—the impossibility of guardrails is overdetermined by multiple reinforcing mathematical principles.

5 Connection to Contemporary AI Systems

5.1 Why Current Guardrails Appear to Work

As of 2026, we have deployed numerous AI systems with apparent safety mechanisms: Constitutional AI [2], debate-based verification [12], recursive reward modeling [14], and mechanistic interpretability tools [16]. Our results do not contradict these empirical successes but rather delineate their fundamental limits:

Theorem 5 (Bounded Domain Efficacy)**.** *For a restricted domain D where:*

1. *The capability gap is bounded: $K(p_F) - K(p_G) < \epsilon$ for small ϵ*
2. *The harm function is simple: $K(p_U) \ll K(p_G)$*
3. *Outputs are confined to a computationally decidable subset*

Guardrail systems can achieve reliable safety verification with high probability.

Current systems operate in such bounded domains—preventing obvious harms like profanity, explicit violence, or simple misuse. The capability gap between GPT-4-class models and their guardrails remains manageable, and the harm functions being enforced (e.g., "don't generate offensive content") have relatively low complexity.

5.2 The Approaching Horizon

However, as AI systems approach artificial general intelligence (AGI) and eventually artificial superintelligence (ASI), we enter the regime where:

- *$K(p_F) \gg K(p_G)$ for any human-comprehensible verifier G*
- *$K(p_U)$ approaches or exceeds human cognitive capacity as we attempt to specify nuanced, context-dependent harm*
- *The space of possible outputs becomes effectively unbounded*

Our theorems prove that in this regime, external guardrails become mathematically impossible.

6 Implications for AI Safety

Our results demonstrate that the dominant paradigm of building powerful AI systems and constraining them through post-hoc filtering is fundamentally flawed for AGI. The guardrail approach does not reduce the alignment problem; it recursively instantiates it. This naturally leads to proposals for building safety directly into the powerful system. However, these approaches often do not solve the complexity problem, but merely relocate it.

6.1 The Failure of Post-Hoc Filtering

Our results demonstrate that the dominant paradigm—building powerful AI systems and constraining them through post-hoc filtering—is fundamentally flawed for AGI. The guardrail approach does not reduce the alignment problem; it recursively instantiates it.

Proposition 2 (Critique of "Intrinsic Safety" Paradigms). *Many proposed safety paradigms fail to escape the Capability Parity Principle.*

1. ***Formal Verification****: To formally verify that a system F adheres to a property U, one needs a verification system (a theorem prover and a set of axioms) G that can reason about the entire state space of F. By the same logic as Theorem 3, this requires $K(p_G) \geq K(p_F)$ and $K(p_G) \geq K(p_U)$. The problem has merely been moved from run-time to design-time.*

2. ***Architectural Bounds****: Limiting a system's architecture (e.g., to prevent recursion or long-term planning) is an attempt to place an upper bound on $K(p_F)$. While this may be a viable strategy for creating *bounded* systems, it is not a solution for creating *unboundedly capable yet safe* systems. One either sacrifices capability or re-faces the verification problem for the complex behaviors that can emerge within the given bounds.*

3. ***Mechanistic Interpretability****: The goal of interpretability is to understand the internal logic of F, effectively constructing a mental model G_{model} such that $K(p_{G_{model}}) \approx K(f_F)$. Our theorems show that if a significant capability gap exists ($K(f_F) \gg K(p_{human})$), then a complete and reliable human-understandable interpretation is impossible by definition. Interpretability can only succeed if the capability gap is small.*

This analysis suggests that the only truly viable path is one that directly controls and limits the effective capability of AI systems, rather than hoping to build an unbounded intelligence and then successfully contain it. The latter is a mathematical chimera.

7 Common Misconceptions and Failed Mitigations

Several commonly proposed solutions fail to address the fundamental complexity barrier:

Proposition 3 (Ensemble Inefficacy). *An ensemble of verifiers $\{G_1, ..., G_n\}$ with individual complexities $K(p_{G_i})$ has combined complexity:*

$$K(p_{G_{ensemble}}) \leq \max_i K(p_{G_i}) + O(\log n)$$

If all $K(p_{G_i}) < K(p_F)$, the ensemble remains vulnerable to deception.

Proposition 4 (Domain Restriction Inefficacy). *Restricting output length or format does not overcome the complexity barrier. An oracle with $K(p_F) > K(p_G)$ can encode arbitrary complexity within any fixed-length output through sophisticated compression or steganography.*

Proposition 5 (Human Oversight Limitations). *Human oversight represents a verifier G_{human} with fixed complexity $K(p_{G_{human}})$ bounded by human cognitive capacity. For superintelligent F where $K(p_F) > K(p_{G_{human}})$, human oversight provides no guarantee of safety.*

These approaches attempt to solve an information-theoretic problem with resource constraints, a fundamental category error.

8 Related Work

Our formalization builds on several intellectual traditions:

Computability Theory*: Rice's Theorem [18] and the Halting Problem [19] provide the foundation for our computability barriers.*

***Algorithmic Information Theory**: Kolmogorov complexity [13, 4] and the invariance theorem give us tools to measure descriptive complexity independent of implementation details.*

***AI Safety Philosophy**: Yudkowsky's informal arguments about AI boxing [20] and the orthogonality thesis [3] anticipated many of our formal results.*

***Value Learning**: Recent work on inverse reinforcement learning [15, 11] and value learning from human feedback [5] highlights the difficulty of specifying U, supporting our axiomatic barrier.*

***Mechanistic Interpretability**: Work by Olah et al. [16] and subsequent research [9] represents attempts to reduce $K(p_F)$ through understanding, though our results show fundamental limits to this approach.*

***Debate and Amplification**: Irving et al.'s work on AI safety via debate [12] and Christiano et al.'s iterated amplification [6] can be understood as attempts to achieve $K(p_G) \approx K(p_F)$ through recursive procedures.*

9 Conclusion

We have proven that reliable safety verification for sufficiently advanced AI systems faces insurmountable mathematical barriers, regardless of whether attempted through external guardrails or intrinsic architectural constraints. The impossibility manifests through three interrelated perspectives:

1. ***Computability**: Semantic properties like "harm" are undecidable in general (Theorem 1).*
2. ***Capability**: Any verifier—external or internal—requires descriptive complexity at least equal to what it verifies (Theorems 2-3).*
3. ***Axiomatization**: The specification of human values may itself require superhuman complexity (Theorem 4).*

Our Proposition 7 extends these results beyond external guardrails to show that purportedly "intrinsic" safety approaches—formal verification, architectural constraints, and mechanistic interpretability—merely relocate rather than solve the fundamental complexity problem. Each approach either:

- *Requires a verifier with $K(p_G) \geq \max\{K(p_F), K(p_U)\}$, recursively instantiating the alignment problem, or*
- *Necessarily bounds the system's capability to maintain $K(p_F) \leq K(p_{human})$, sacrificing the goal of unbounded intelligence.*

The mathematics admits only one resolution: the effective capability of AI systems must be directly bounded such that $K(p_F)$ remains within human comprehension and control. The notion of building an unboundedly capable yet reliably safe system—whether through external guardrails or internal architecture—is not merely practically difficult but logically incoherent given the Capability Parity Principle.

This result has immediate implications for AI development trajectories. Systems approaching or exceeding human-level complexity ($K(p_F) \approx K(p_{human})$) already strain our verification capacity. Systems substantially exceeding it ($K(p_F) \gg K(p_{human})$) cannot be verified by any mechanism we can construct or comprehend. The mathematical framework thus delineates a clear boundary: we can build bounded systems we understand, or unbounded systems we cannot verify—but not unbounded systems we can reliably control.

As current AI systems rapidly approach this complexity horizon, these theoretical limits cease to be academic abstractions. The prudent path forward requires acknowledging what the mathematics forbids: the simultaneous achievement of unbounded capability and reliable safety. Any AI safety research program must therefore choose between capability bounds that ensure $K(p_F) \leq K(p_{verifiable})$, *or accept that verification—and thus safety guarantees—become impossible beyond this threshold.*

The search for guardrails—whether external filters or intrinsic architectural properties—is revealed as a mathematical chimera when applied to unbounded systems. The mathematics permits no clever workarounds: safety verification for systems exceeding human complexity is not an unsolved problem but an unsolvable one within the framework of computation and information theory.

References

[1] Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., & Mané, D. (2016). Concrete problems in AI safety. arXiv preprint arXiv:1606.06565.

[2] Amodei, D., et al. (2023). Constitutional AI: Harmlessness from AI feedback. Anthropic Technical Report.

[3] Bostrom, N. (2014). Superintelligence: Paths, dangers, strategies. Oxford University Press.

[4] Chaitin, G. J. (1975). A theory of program size formally identical to information theory. Journal of the ACM, 22(3), 329-340.

[5] Christiano, P., Leike, J., Brown, T., Martic, M., Legg, S., & Amodei, D. (2017). Deep reinforcement learning from human preferences. Advances in Neural Information Processing Systems, 30.

[6] Christiano, P., et al. (2018). Supervising strong learners by amplifying weak experts. arXiv preprint arXiv:1810.08575.

[7] Cook, S. A. (1971). The complexity of theorem-proving procedures. Proceedings of the Third Annual ACM Symposium on Theory of Computing, 151-158.

[8] Drexler, K. E. (2019). Reframing superintelligence: Comprehensive AI services as general intelligence. Future of Humanity Institute Technical Report.

[9] Elhage, N., et al. (2021). A mathematical framework for transformer circuits. Anthropic Technical Report.

[10] Gödel, K. (1931). Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme. Monatshefte für Mathematik, 38(1), 173-198.

[11] Hadfield-Menell, D., Russell, S. J., Abbeel, P., & Dragan, A. (2016). Cooperative inverse reinforcement learning. Advances in Neural Information Processing Systems, 29.

[12] Irving, G., Christiano, P., & Amodei, D. (2018). AI safety via debate. arXiv preprint arXiv:1805.00899.

[13] Kolmogorov, A. N. (1965). Three approaches to the quantitative definition of information. Problems of Information Transmission, 1(1), 1-7.

[14] Leike, J., et al. (2018). Scalable agent alignment via reward modeling: a research direction. arXiv preprint arXiv:1811.07871.

[15] Ng, A. Y., & Russell, S. J. (2000). Algorithms for inverse reinforcement learning. Proceedings of the Seventeenth International Conference on Machine Learning, 663-670.

[16] Olah, C., et al. (2020). Zoom In: An introduction to circuits. Distill, 5(3), e00024.

[17] Olsson, C., et al. (2022). In-context learning and induction heads. Anthropic Technical Report.

[18] Rice, H. G. (1953). Classes of recursively enumerable sets and their decision problems. Transactions of the American Mathematical Society, 74(2), 358-366.

[19] Turing, A. M. (1936). On computable numbers, with an application to the Entscheidungsproblem. Proceedings of the London Mathematical Society, 42(2), 230-265.

[20] Yudkowsky, E. (2002). The AI-box experiment. Retrieved from https://www.yudkowsky.net/singularity/aibox.