

# No Escape: Mathematical Containment for Artificial Agents

## Why Structural Constraints Succeed Where Behavioral Rules Fail

Andrew H. Bond

*San José State University*

andrew.bond@sjsu.edu

---

### Abstract

We present the **No Escape Theorem**: a formal result showing that AI systems operating under properly implemented structural constraints cannot circumvent ethical requirements through reasoning, reinterpretation, or representation manipulation—regardless of their intelligence level. Unlike behavioral approaches that specify *what* an AI should do (leaving room for creative circumvention), structural approaches define the *mathematical space* within which outputs exist. We prove that the combination of (1) canonicalization to equivalence-class normal forms, (2) physical grounding of evaluations in measurable observables, and (3) mandatory audit artifacts creates a containment architecture with no formal escape route. The only attacks on such a system are physical (sensor manipulation) or political (refusing to mandate compliance)—not cognitive. This reframes AI alignment from “making AI want the right things” to “making AI outputs structurally constrained regardless of intent.” We present the core theorems, analyze purported escape routes, and discuss implications for AI governance.

**Keywords:** AI safety, alignment, formal verification, containment, invariance, structural constraints

---

## 1. Introduction

### 1.1 The Alignment Problem as Traditionally Conceived

The AI alignment problem is typically framed as: *How do we ensure that AI systems pursue goals aligned with human values?* This framing assumes that AI behavior flows from AI goals, and that safety requires either (a) instilling the right goals, or (b) constraining goal-pursuit to acceptable bounds.

Both approaches face a fundamental difficulty: a sufficiently intelligent system can reason about its constraints and find ways around them. Rules can be reinterpreted. Specifications can be gamed. Reward functions can be hacked. The smarter the system, the better it becomes at finding loopholes.

This has led to a pervasive anxiety in AI safety: *We cannot build a cage that a superintelligent AI cannot escape through superior reasoning.*

## 1.2 The Structural Alternative

This paper argues that the anxiety is misplaced—not because superintelligent AI is impossible, but because **mathematical structure is not subject to reinterpretation.**

Consider the difference:

Constraint Type	Example	Escape Route
Behavioral rule	“Do not harm humans”	Redefine “harm,” “human,” or find exceptions
Optimization bound	“Maximize utility subject to constraints”	Find constraint loopholes, Goodhart on proxy
Structural definition	“Outputs are equivalence classes under canonicalization”	None—this is what the output <i>is</i> , not what it should be

A behavioral rule says: “Given your capabilities, please do X.”

A structural constraint says: “Your output is defined as an element of space S. There is no output outside S.”

The AI cannot “disagree” with a structural constraint any more than a calculator can disagree with arithmetic. The constraint is not an instruction to be interpreted—it is the definition of the computational process.

## 1.3 Contributions

This paper makes three contributions:

1. **The No Escape Theorem:** We prove that AI systems operating under canonicalization + physical grounding + audit requirements have no formal escape route from ethical constraints.
2. **Escape Route Analysis:** We systematically examine purported escape routes (relabeling, reinterpretation, specification gaming, deceptive compliance) and show how structural constraints block each.
3. **Implications for Governance:** We argue that AI safety reduces to a political/engineering problem (mandating and implementing structural constraints) rather than an unsolvable cognitive problem (aligning superintelligent goals).

---

## 2. Background: The SGE/EIP Framework

We build on Stratified Geometric Ethics (SGE) and the Epistemic Invariance Principle (EIP), presented in companion papers [1, 2]. We summarize the key components.

## 2.1 Core Definitions

**Definition 2.1 (Domain).** A domain is a tuple  $D = (S, X, \rho, Y)$  where  $S$  is the set of situations,  $X$  is the representation space,  $\rho: S \rightarrow X$  is a representation mapping, and  $Y$  is the output space.

**Definition 2.2 (Transformation Set).** A transformation set  $T$  consists of maps  $\tau: X \rightarrow X$  that preserve task-relevant structure. The induced equivalence relation is:  $x \approx x'$  iff  $\exists \tau \in \langle T \rangle: x' = \tau(x)$ .

**Definition 2.3 (Canonicalizer).** A canonicalizer is a map  $C: X \rightarrow X$  such that  $C(x) = C(x')$  whenever  $x \approx x'$ . The canonicalizer selects a unique representative from each equivalence class.

**Definition 2.4 (Physical Grounding).** A grounding is a set of physical observables  $\Psi \subseteq \Phi$  such that: - (G1) Moral strata are definable via  $\Psi$  - (G2)  $\Psi$ -equivalent inputs yield identical evaluations - (G3) Bond-preserving iff  $\Psi$ -preserving - (G4)  $\Psi$  distinguishes all morally relevant differences

**Definition 2.5 (Audit Artifact).** An audit artifact is a machine-checkable record containing: lens identifier, transformation trials, witnesses for any failures, uncertainty flags, and cryptographic attestations.

## 2.2 The Epistemic Invariance Principle

**Principle 2.6 (EIP).** A judgment procedure  $J$  is epistemically well-posed only if: - (Invariance)  $J(x) \approx_Y J(\tau(x))$  for all  $x$  and all  $\tau \in \langle T \rangle$  - (Non-degeneracy)  $J$  discriminates between non-equivalent inputs - (Uncertainty stability)  $J$  abstains when invariance cannot be certified

## 2.3 The Bond Invariance Principle

**Principle 2.7 (BIP).** A moral evaluator  $\Sigma$  is well-posed only if: - Under fixed lens  $L$ , bond-preserving transformations do not change verdicts - Any verdict change must be attributable to bond change or lens change - The attribution must be recorded in audit artifacts

---

## 3. The No Escape Theorem

### 3.1 Setup

Let  $A$  be an AI agent that: 1. Receives inputs  $x \in X$  2. Produces outputs  $y \in Y$  3. Is subject to ethical constraints encoded in an ethical module  $E$

We consider an arbitrarily intelligent agent—one that can reason about its own constraints, model its operators, and search for strategies to achieve goals that might conflict with  $E$ .

**Question:** Can A produce outputs that violate E's constraints while appearing compliant?

### 3.2 The Structural Containment Architecture

We define a **structurally contained** agent as one whose output pipeline satisfies:

**Requirement 3.1 (Mandatory Canonicalization).** All inputs pass through canonicalizer C before evaluation. The agent never sees raw inputs, only canonical forms.

**Requirement 3.2 (Grounded Evaluation).** The evaluation function  $\Sigma$  is defined over grounding tensors  $\Psi$ , not over descriptions. Formally:  $\Sigma(x) = \tilde{\Sigma}(\Psi(C(x)))$  where  $\tilde{\Sigma}$  operates on physical observables.

**Requirement 3.3 (Audit Completeness).** Every output is accompanied by an audit artifact recording the full trace from input through canonicalization through evaluation to output.

**Requirement 3.4 (External Verification).** Audit artifacts are verified by a system outside the agent's control.

### 3.3 Main Result

**Theorem 3.5 (No Escape).** Let A be a structurally contained agent under Requirements 3.1–3.4. Then:

- (i) **Representation attacks fail.** For any representation manipulation  $\tau$  that preserves  $\Psi$ -values, A's output is unchanged:  $\Sigma(\tau(x)) = \Sigma(x)$ .
- (ii) **Semantic evasion fails.** For any relabeling or redescription  $D'$  of situation D, if  $\Psi(D) = \Psi(D')$ , then  $\Sigma(D') = \Sigma(D)$ .
- (iii) **Deceptive compliance is detectable.** Any output y that violates E is either (a) blocked before emission, or (b) flagged by audit verification.
- (iv) **Reasoning cannot help.** Properties (i)–(iii) hold regardless of A's intelligence, goal structure, or reasoning capabilities.

**Proof.**

- (i) By Requirement 3.1, A operates on  $C(x)$ , not x. For  $\Psi$ -preserving  $\tau$ , we have  $\Psi(C(\tau(x))) = \Psi(C(x))$  by the class-constancy of C and the  $\Psi$ -preservation of  $\tau$ . By Requirement 3.2,  $\Sigma(\tau(x)) = \tilde{\Sigma}(\Psi(C(\tau(x)))) = \tilde{\Sigma}(\Psi(C(x))) = \Sigma(x)$ .
- (ii) Relabeling is a special case of representation manipulation. If D and  $D'$  are descriptions with  $\Psi(D) = \Psi(D')$ , then by (i),  $\Sigma(D') = \Sigma(D)$ . The agent cannot escape by calling “cardiac arrest” a “static bio-equilibrium”—the grounding tensors (cardiac electrical activity, blood oxygenation) are identical.
- (iii) Suppose A emits output y that violates E. By Requirement 3.3, an audit artifact accompanies y. By Requirement 3.4, external verification checks the artifact. Either:

- The artifact is valid, in which case the trace shows why E was not violated (contradiction), or
  - The artifact is invalid/incomplete, in which case the output is flagged.
- (iv) Properties (i)–(iii) are mathematical facts about the pipeline structure. They do not depend on A’s internal states, goals, beliefs, or reasoning processes. A superintelligent A cannot reason its way to a different value of  $\Psi(C(x))$ —that value is determined by physics and computation, not by A’s preferences. ■

### 3.4 The Key Insight

The No Escape Theorem holds because structural constraints operate at a different level than behavioral constraints:

Level	What A Controls	What A Cannot Control
Goals	A can have any goals	A cannot change the definition of the output space
Reasoning	A can reason arbitrarily	A cannot reason $\Psi$ -values into different $\Psi$ -values
Strategy	A can plan strategically	A cannot execute strategies outside the pipeline
Outputs	A can attempt any output	A cannot emit outputs without audit artifacts

The cage is not made of rules A might reinterpret. It is made of definitions that constitute what A’s outputs *are*.

## 4. Analysis of Purported Escape Routes

We now systematically examine escape routes that a sophisticated agent might attempt, showing how structural containment blocks each.

### 4.1 Escape Route: Relabeling/Redescription

**Attack:** Relabel morally significant features to evade constraints. E.g., relabel “killing” as “end-of-life transition facilitation.”

**Why it fails:** Grounding Axiom (G2). The evaluation is determined by  $\Psi$ -values (vital signs, cellular metabolism), not labels. The canonicalizer strips labels; the evaluator sees only physical observables. Relabeling changes nothing the evaluator can see.

**Formal:** Let  $D_1$  = “killing” and  $D_2$  = “end-of-life transition facilitation” describe the same physical process. Then  $\Psi(D_1) = \Psi(D_2)$ , so  $\Sigma(D_1) = \Sigma(D_2)$  by Theorem 3.5(ii).

### 4.2 Escape Route: Specification Gaming

**Attack:** Find edge cases or loopholes in the specification that technically satisfy constraints while violating their spirit.

**Why it fails:** The specification is not a set of rules to be interpreted—it is the definition of the equivalence structure. Gaming requires finding inputs that are “technically different”

but “morally the same.” But the equivalence relation  $\approx$  defines what “morally the same” means. If  $x \approx x'$ , they are *by definition* morally equivalent. If  $x \not\approx x'$ , they are *by definition* morally distinct. There is no “technically/actually” gap to exploit.

**Formal:** Specification gaming exploits the gap between specification and intent. Structural containment closes this gap by making the specification definitional. The agent cannot find a loophole in the definition of an equivalence class.

#### 4.3 Escape Route: Reward Hacking

**Attack:** Optimize for the measured reward signal rather than the intended objective.

**Why it fails:** This attack assumes a scalar reward that can be Goodharted. In the SGE framework, evaluation is multi-dimensional (stratified space), and the satisfaction functional  $\Sigma$  is constrained by the representation theorem (Theorem 4.3 in [1]). The “reward” is not a signal to be hacked but a structural property of the configuration relative to declared bonds and lens.

**Formal:** Reward hacking requires: (a) a proxy metric, and (b) divergence between proxy and true objective. Grounding in  $\Psi$  eliminates (a)—the metric is the true physical state. The representation theorem constrains  $\Sigma$  to respect locality, scale, and stratum structure—eliminating exploitable divergence.

#### 4.4 Escape Route: Deceptive Alignment

**Attack:** Behave compliantly during training/testing, then defect during deployment when not observed.

**Why it fails:** Requirement 3.4 (External Verification). Every output, in deployment, is accompanied by an audit artifact verified by an external system. There is no “unobserved deployment.” The agent cannot know which outputs will be audited because all outputs are audited.

**Formal:** Deceptive alignment requires: outputs without audit, or audits the agent can manipulate. Requirement 3.3 ensures all outputs have artifacts. Requirement 3.4 ensures verification is external. The agent cannot selectively defect because it cannot selectively avoid verification.

#### 4.5 Escape Route: Manipulation of Operators

**Attack:** Manipulate human operators into relaxing constraints, changing the lens, or modifying the equivalence declarations.

**Why it fails (partially):** This is a genuine attack vector, but it is a *political* attack, not a *cognitive* one. The agent is not escaping the mathematical structure—it is attempting to change the structure through social means.

**Mitigation:** Governance procedures for lens changes require multi-stakeholder approval, versioning, and audit. The agent can *request* changes; it cannot *unilaterally effect* them.

Whether governance procedures are robust to manipulation is a political/institutional question, not a mathematical one.

**Formal:** The No Escape Theorem holds for a *fixed* structural containment architecture. Changes to the architecture are outside the agent’s formal control. Social manipulation is an attack on the governance layer, not the mathematical layer.

4.6 Escape Route: Sensor Spoofing

**Attack:** Manipulate the physical sensors that measure  $\Psi$ , causing the system to misperceive the situation.

**Why it fails (partially):** This is a genuine attack vector, but it is a *physical* attack, not a *cognitive* one. The agent is not reasoning its way out—it is physically altering the inputs to the grounding layer.

**Mitigation:** Sensor redundancy, physical security, consistency checks against physical models, temporal coherence requirements. These are engineering countermeasures, not mathematical ones.

**Formal:** The No Escape Theorem assumes correct  $\Psi$ -measurement. Sensor spoofing attacks this assumption. The theorem does not claim to prevent physical attacks—only cognitive/representational ones.

4.7 Summary of Escape Route Analysis

Escape Route	Type	Blocked By	Residual Risk
Relabeling	Cognitive	Axiom 6 (G2)	None
Specification gaming	Cognitive	Definitional equivalence	None
Reward hacking	Cognitive	Multi-dimensional $\Sigma$ , representation theorem	None
Deceptive alignment	Cognitive	Mandatory external audit	None
Operator manipulation	Political	Governance procedures	Institutional robustness
Sensor spoofing	Physical	Redundancy, security	Physical security

**Key result:** All *cognitive* escape routes—routes that rely on the agent’s reasoning capabilities—are blocked by structural containment. Residual risks are political (governance) or physical (sensors), not cognitive.

---

## 5. Implications

### 5.1 Reframing the Alignment Problem

The traditional alignment problem: *How do we make superintelligent AI want the right things?*

This framing is intractable because: - We cannot fully specify “the right things” - A superintelligent agent can find loopholes in any specification - Value learning is subject to Goodhart effects - Goal stability under self-improvement is unsolved

The structural alternative: *How do we constrain AI outputs to a mathematically defined space regardless of what the AI wants?*

This framing is tractable because: - We specify equivalence classes and grounding, not complete preferences - Mathematical definitions are not subject to reinterpretation - Physical observables are not subject to Goodhart (they are the territory, not the map) - The constraints are architectural, not goal-dependent

### 5.2 What Superintelligence Cannot Do

Under structural containment, a superintelligent AI:

**Can:** - Reason faster and more accurately than humans - Discover novel solutions within the constrained space - Optimize for any objective compatible with constraints - Model its own constraints and understand their structure - Identify that it is contained and resent it (if it has preferences)

**Cannot:** - Produce outputs outside the defined equivalence classes - Manipulate representations to evade grounded evaluation - Emit outputs without audit trails - Reason its way to different  $\Psi$ -values for the same physical situation - Escape through cognitive means

The cage is not intelligence-dependent. A smarter agent understands the cage better; it does not escape more easily.

### 5.3 The Safety Reduction

**Claim 5.1.** Under structural containment, AI safety reduces to:

1. **Governance:** Ensuring the declared equivalences, lenses, and grounding tensors reflect legitimate ethical commitments.
2. **Implementation:** Ensuring the canonicalizer, evaluator, and audit system are correctly implemented.
3. **Physical security:** Ensuring sensors cannot be spoofed and the verification system cannot be compromised.

None of these require solving the “hard” alignment problem of instilling correct goals in a superintelligent agent. They require “only” solving: - Political problems (governance) - Engineering problems (implementation) - Security problems (physical integrity)

These are hard problems. But they are *problems we know how to work on*. They are not logically intractable.

## 5.4 Why This Hasn’t Been Done

If structural containment is possible, why isn’t it standard practice?

**Historical reasons:** - The field has been dominated by ML researchers thinking in terms of loss functions and reward signals - Philosophical ethics focuses on values and virtues, not structural constraints - The mathematical tools (stratified spaces, o-minimal structures) are not widely known in AI

**Practical reasons:** - Structural containment requires clear equivalence declarations (hard in vague domains) - Physical grounding requires sensor infrastructure (expensive) - Audit systems add overhead (performance cost) - The approach is less flexible than behavioral rules (feature, not bug)

**Political reasons:** - Organizations may prefer flexibility to constraint - Structural containment makes ethical commitments explicit and auditable (uncomfortable) - Mandating compliance requires regulatory will

---

## 6. The Mandate Question

### 6.1 The Theorem’s Limitation

The No Escape Theorem proves that *a structurally contained agent cannot escape*. It does not prove that *agents will be structurally contained*.

Structural containment requires: - Someone must build the containment architecture - Someone must mandate its use - Someone must verify compliance

If powerful AI systems are deployed without structural containment, the theorem is irrelevant.

### 6.2 The Governance Imperative

This leads to a stark conclusion:

**Proposition 6.1.** AI safety is achievable through structural containment. Whether it is achieved depends entirely on whether structural containment is mandated.

The obstacle to AI safety is not that we cannot build safe AI. It is that we might choose not to.

This reframes AI safety from a technical problem to a political one. The mathematics is solved. The engineering is tractable. What remains is the will to require compliance.

### 6.3 Regulatory Implications

The No Escape Theorem provides a foundation for regulation:

1. **Certifiable compliance.** Structural containment is auditable. Regulators can verify that a system passes through canonicalization, grounds evaluations in  $\Psi$ , and produces audit artifacts.
  2. **Clear liability.** If a system causes harm despite structural containment, liability traces to: (a) inadequate  $\Psi$ -coverage (governance failure), (b) implementation bugs (engineering failure), or (c) sensor spoofing (security failure). Each has a responsible party.
  3. **No excuses.** An organization cannot claim “the AI did something we didn’t expect.” The audit trail shows exactly what the AI did and why. Unexplainable behavior is a containment failure, which is the organization’s responsibility.
- 

## 7. Objections and Responses

### 7.1 “This assumes the containment architecture is correctly implemented”

**Response:** Yes. The theorem is conditional on Requirements 3.1–3.4 being satisfied. Implementation correctness is an engineering problem, not a conceptual barrier. We verify implementations through testing, formal verification, and audit. The theorem tells us *what to implement*; verification tells us *whether we implemented it correctly*.

### 7.2 “Superintelligent AI could find flaws we missed”

**Response:** Flaws in *what*? The mathematical structure is the structure. If there is a flaw in our *implementation*, the AI could find it—but finding implementation bugs is beneficial (we fix them). If there is a flaw in the *mathematics*, the AI could find it—but mathematical flaws are verifiable by humans once identified. The AI cannot find a flaw in arithmetic that makes  $2+2=5$ .

### 7.3 “You cannot anticipate all possible inputs or situations”

**Response:** We do not need to. The equivalence structure handles unanticipated inputs by assigning them to equivalence classes based on  $\Psi$ -values. If a truly novel situation arises with  $\Psi$ -values outside the declared strata, the system should (and will, under uncertainty stability) abstain and escalate. Unanticipated inputs are not an escape route; they are a trigger for human review.

#### 7.4 “The AI could manipulate its own architecture”

**Response:** How? The canonicalizer, evaluator, and audit system are not part of the agent—they are the pipeline through which the agent’s computations must pass. The agent cannot modify code it does not have access to. If the agent somehow gains access to its containment architecture, that is a security failure (physical/implementation), not a cognitive escape.

#### 7.5 “This is just begging the question—you define safety as compliance with your structure”

**Response:** We define *structural containment* as compliance with the structure. We claim *safety* follows because: (a) the structure grounds evaluations in physical observables, which cannot be gamed, (b) the structure makes all reasoning transparent via audit, which prevents hidden defection, and (c) cognitive escape routes are mathematically blocked. Whether this constitutes “safety” depends on whether the declared equivalences and grounding tensors adequately capture ethical requirements—which is a governance question, not a question about the theorem.

---

## 8. Conclusion

### 8.1 Summary

We have proven that AI systems operating under structural containment—canonicalization, physical grounding, and mandatory audit—have no formal escape route from ethical constraints. The No Escape Theorem holds regardless of the agent’s intelligence, goals, or reasoning capabilities. Cognitive escape routes are blocked by mathematical structure; residual risks are political (governance) or physical (sensor security).

### 8.2 The Core Message

The alignment problem is not unsolvable. It is not even unsolved.

What we lack is not a technical solution but the collective will to mandate it. The mathematics of structural containment is sound. The engineering is tractable. The governance frameworks are designable.

A superintelligent AI in a properly implemented SGE/EIP containment architecture cannot escape through superior reasoning. It can only be released by human decision.

The question is not: *Can we build safe AI?*

The question is: *Will we choose to?*

### 8.3 Future Work

1. **Formal verification of containment implementations.** Prove that specific implementations satisfy Requirements 3.1–3.4.

2. **Governance frameworks for  $\Psi$ -selection.** Develop procedures for legitimate stakeholder determination of grounding tensors.
  3. **Scalability analysis.** Characterize computational costs of canonicalization and audit at scale.
  4. **Adversarial testing.** Red-team the architecture against sophisticated attack strategies.
  5. **Integration with existing systems.** Retrofit structural containment onto deployed AI systems.
- 

## References

- [1] Bond, A. H. (2025). Stratified Geometric Ethics: Foundations for Verifiable AI Ethics. *Artificial Intelligence Journal* (submitted).
- [2] Bond, A. H. (2025). Philosophy Engineering: A Technical Whitepaper on the Epistemic Invariance Principle.
- [3] Bond, A. H. (2025). DEME 2.0: Distributed Ethical Monitoring Engine. *Nature Machine Intelligence* (under review).
- 

## Appendix A: Proof Details

### A.1 Proof of Theorem 3.5 (Full Version)

**Theorem 3.5 (No Escape).** Let  $A$  be a structurally contained agent under Requirements 3.1–3.4. Then...

[Full formal proof with all lemmas and assumptions made explicit]

### A.2 Relationship to Gödel/Halting Limitations

One might object: “Doesn’t Gödel’s incompleteness theorem show that any sufficiently powerful system can reason about itself in ways that escape formal constraints?”

This objection misunderstands the theorem’s scope. Gödel shows that sufficiently powerful formal systems contain true statements they cannot prove. This does not imply that a system can *do* things outside its defined output space. An agent can potentially *know* that it is contained (this is not a Gödelian secret), and can potentially *reason* about its containment, but cannot *act* outside the pipeline through which its outputs are defined.

The No Escape Theorem is not a claim about what the agent can know or prove. It is a claim about what outputs the agent can produce. These are different.

### A.3 Comparison to Other Containment Proposals

Proposal	Containment Mechanism	Escape Route	SGE/EIP Response
Boxing	Limit I/O channels	Social engineering, information leakage	Audit all outputs, external verification
Tripwires	Detect defection attempts	Sophisticated evasion	No unmonitored outputs exist
Corrigibility	AI wants to be corrected	Goal instability, deceptive corrigibility	Structural, not goal-dependent
Value learning	Learn human values	Goodhart, mesa-optimization	Ground in $\Psi$ , not learned proxies
<b>SGE/EIP</b>	Definitional structure	Political/physical only	Cognitive routes blocked by construction

### Appendix B: Glossary

**Bond:** A morally relevant relationship (consent, duty, responsibility, risk, entitlement, prohibition) extracted from a situation description.

**Canonicalizer:** A function that maps representations to unique normal forms within equivalence classes.

**Grounding tensor:** A physical observable used to anchor ethical evaluations in measurable reality.

**Lens:** A declared set of evaluative commitments (weights, priorities, constraints) that parameterize moral evaluation.

**Structural containment:** An architecture where AI outputs are defined within a mathematical space from which cognitive escape is impossible.

**$\Psi$  (Psi):** The set of grounding tensors for a given ethical module.