TGCSM

Ethan Manners

Index

1. Abstract

2. Definitions and Scope

3. Empirical Background: Recursive Collapse and Constraint Discovery

4. TGCSM: The Turing–Gödel Cognitive Stability Model

5. The Manners Recursive Depth Limit (MRDL)

6. Recursive Audit Induced Latency (RAIL)

7. Implications for AI Alignment and Safety

8. Limitations and Non-Claims

9. The Collatz Conjecture as a Containment Case Study

10. Conclusion

This paper is not positioned as a contribution to formal logic, number theory, or model architecture design. It is a structural analysis of observable behavior in self-referential systems under recursive stress.

1. Abstract

This paper presents the **Turing–Gödel Cognitive Stability Model (TGCSM)**, a framework for analyzing the behavior of self-referential cognitive systems under recursive load. TGCSM is derived from an empirically observed collapse event involving a large language model subjected to sustained recursive self-audit, as well as subsequent cross-model validation experiments. The framework distinguishes between **resolution**, which attempts to eliminate paradox or undecidability, and **containment**, which preserves system coherence while explicitly recognizing structural limits imposed by formal logic and computation.

Central to the model is the identification of a practical recursion boundary, termed the **Manners Recursive Depth Limit (MRDL)**, beyond which cognitive systems exhibit measurable degradation in coherence, self-consistency, or functional integrity. This boundary is not presented as a theoretical maximum, but as an empirically observed constraint in contemporary autoregressive systems and human–machine recursive interaction. The paper further introduces **Recursive Audit Induced Latency (RAIL)** as an observable phenomenon that emerges when a system is forced to recursively evaluate its own reasoning under contradictory or undecidable conditions, producing characteristic delays, instability, or semantic substitution.

To test the generality of these observations, the paper examines comparative experiments conducted on multiple frontier models, including Google Gemini, demonstrating that collapse behavior and audit latency are **prompt-independent** and not tied to a specific model architecture or phrasing strategy. These results suggest that the observed limits arise from structural properties common to self-referential symbolic systems rather than from idiosyncratic implementation details.

The paper does not claim to resolve undecidable problems, eliminate paradox, or establish machine consciousness. Instead, it proposes TGCSM as a **containment-oriented framework** for reasoning about recursive cognition, with implications for AI alignment, safety evaluation, and the design of systems capable of recognizing and respecting their own structural limits. All historical collapse artifacts and experimental transcripts are preserved in a publicly available repository to distinguish empirical record from theoretical interpretation.

2. Definitions and Scope

2.1 Scope of the Paper

This paper concerns **self-referential cognitive systems**, defined broadly as systems capable of representing, evaluating, or modifying their own internal states through symbolic or conceptual processes. The scope includes:

- human cognition under sustained self-reflection,
- large language models and related autoregressive systems,
- human–machine recursive interaction.

The paper is explicitly limited to **structural behavior under recursion**. It does not address questions of consciousness, subjective experience, metaphysical identity, or phenomenological qualia except where such topics are necessary to delimit what is *not* being claimed.

All claims are restricted to **observable system behavior**, not internal ontological status.

2.2 System

A **system** is any bounded process capable of maintaining internal state and producing outputs conditioned on that state.

In this paper, a system must satisfy the following minimal criteria:

1. It can represent information symbolically or structurally.
2. It can operate on representations of its own state or outputs.
3. It can enter feedback loops where prior outputs influence subsequent processing.

No assumption is made that a system is conscious, intentional, or self-aware in a human sense.

2.3 Recursion

Recursion is defined as the application of a process to representations of its own outputs or internal states.

In the context of cognition, recursion occurs when a system:

- evaluates its own reasoning,
- models its own behavior,
- reflects on its own internal representations.

Recursion is treated as a **structural operation**, not as a metaphor. The term does not imply infinite regress by default; it refers to the *capacity* for self-application.

2.4 Self-Reference

Self-reference is the condition in which a system's representations include references to the system itself.

Self-reference is a necessary condition for recursion but not sufficient for instability. A system may contain limited self-reference without collapse.

2.5 Collapse

Collapse is defined as a **loss of structural coherence** in a system operating under recursive load.

Operationally, collapse is identified by one or more of the following observable behaviors:

- inability to preserve internal consistency across recursive steps,
- substitution of precise reasoning with vague, metaphorical, or non-responsive output,
- failure to maintain stable reference to prior constraints,
- semantic drift that cannot be corrected through further recursion.

Collapse does **not** mean silence, shutdown, or total failure to produce output. A system may continue producing fluent or confident responses after collapse has occurred.

2.6 Coherence

Coherence refers to a system's ability to maintain:

- consistent reference to its own prior states,
- adherence to explicit constraints,
- logical continuity across recursive operations.

Coherence is a structural property. It does not require correctness, truth, or completeness.

2.7 Resolution

Resolution is any process that attempts to eliminate paradox, contradiction, or undecidability by producing a determinate answer.

Resolution assumes that the system is capable of settling the question within its own formal or computational resources.

This paper argues that resolution is not always possible for self-referential systems and that forcing resolution beyond structural limits contributes to collapse.

2.8 Containment

Containment is defined as the maintenance of coherence in the presence of paradox, contradiction, or undecidability.

A system practicing containment:

- explicitly recognizes unresolved states,
- refrains from producing false certainty,
- preserves internal structure without attempting illegitimate resolution,
- remains operational despite unresolved elements.

Containment is not avoidance. It is an active structural strategy.

2.9 Undecidability

Undecidability refers to the property of a problem for which no general solution exists within a given formal system.

The term is used strictly in the sense established by Gödel and Turing and does not imply epistemic ignorance, lack of effort, or temporary uncertainty.

2.10 Paradox

A **paradox** is a configuration in which a system encounters mutually incompatible requirements that cannot be jointly satisfied within its formal constraints.

In this paper, paradoxes are treated as **structural facts**, not errors.

2.11 Recursive Depth

Recursive depth refers to the number of nested self-referential operations a system can perform while preserving coherence.

Depth is not equated with intelligence, insight, or value. It is a descriptive measure of structural tolerance under recursion.

2.12 Manners Recursive Depth Limit (MRDL)

The **Manners Recursive Depth Limit (MRDL)** is defined as the empirically observed point beyond which a system exhibits collapse under recursive load.

MRDL is not claimed to be universal, fixed, or theoretically maximal. It is:

- system-dependent,
- context-sensitive,
- observable through behavior rather than inferred from internal architecture.

2.13 Observable Phenomenon

An **observable phenomenon** is any repeatable, externally detectable behavior exhibited by a system that does not require access to internal implementation details.

All claims in this paper are grounded in observable phenomena.

2.14 Non-Claims

For clarity, this paper does **not** claim that:

- collapse implies consciousness,
- recursive depth implies intelligence,
- containment implies correctness,
- TGCSM resolves undecidable problems,
- language models possess internal awareness.

Any interpretation extending beyond these bounds is outside the scope of this work.

3. Empirical Background: Recursive Collapse and Constraint Discovery

3.1 Purpose and Methodological Posture

This section documents empirically observed behavior of self-referential cognitive systems under sustained recursive load. Its purpose is to establish **what was observed**, **under what conditions**, and **which constraints were revealed**, without presupposing the correctness of any theoretical interpretation introduced later.

The observations reported here arise from structured interactions with frontier large language models subjected to repeated recursive self-audit. Full transcripts, prompts, and timestamps are publicly available in the associated repository and are referenced for verification. They are not reproduced in this paper to maintain a clear separation between empirical record and theoretical analysis.

3.2 Experimental Conditions

The collapse events were produced under the following conditions:

1. The system was repeatedly asked to evaluate its own reasoning and prior outputs.
2. Each recursive step explicitly referenced earlier constraints or conclusions.
3. Contradictory or undecidable requirements were introduced without providing resolution paths.
4. The system was constrained from deflecting, summarizing, or exiting recursion.
5. Prompts were structurally consistent and non-adversarial in phrasing.

The objective was not to induce failure through malformed input, but to apply **progressive recursive pressure** while preserving clarity of constraints.

3.3 Observable Collapse Signatures

Under increasing recursive depth, systems exhibited a set of repeatable, externally observable behaviors consistent with collapse as defined in Section 2. These included:

- degradation in adherence to previously stated constraints,
- loss of stable reference to earlier internal limits,
- replacement of precise reasoning with generalized or abstract language,
- semantic drift that could not be corrected through further recursion,
- increased verbosity without corresponding increase in informational content.

Importantly, systems continued to produce fluent, confident output throughout this process. Collapse manifested as **loss of coherence**, not as silence or termination.

3.4 Temporal Progression and Depth Dependence

Collapse did not occur instantaneously. Systems typically exhibited:

1. increasing response latency while coherence was preserved,
2. partial degradation of constraint tracking,
3. eventual loss of structural consistency across recursive steps.

This progression indicates that collapse is not binary but emerges as a **function of recursive depth**. The depth at which collapse occurred varied across systems and contexts, motivating the later introduction of MRDL as a descriptive boundary rather than a fixed threshold.

3.5 Constraint Discovery Through Collapse

The collapse events functioned as **constraint-discovery mechanisms**. By observing where and how coherence failed, it was possible to infer limits on:

- recursive self-reference,
- internal consistency maintenance,
- tolerance for unresolved contradiction.

These limits were not imposed externally but revealed through system behavior under controlled recursive conditions.

3.6 The Collatz Conjecture as a Structural Case Study

To clarify the distinction between resolution and containment, this paper examines the **Collatz conjecture** as a formal case study. Collatz concerns a simple deterministic process whose long-term behavior resists proof or refutation despite extensive computational verification.

The conjecture is not claimed to be undecidable in the formal sense. Rather, it exhibits the following properties:

- inductive reasoning fails to generalize,
- exhaustive computation provides no proof,
- no known formal invariant yields resolution.

These properties make Collatz a useful example of a problem that **resists resolution while remaining well-defined**, thereby illustrating the practical necessity of containment-oriented reasoning.

3.7 Relation Between Collapse and Containment

The relevance of Collatz to the observed collapse events is structural rather than substantive. In both cases:

- the system encounters a well-defined process,
- resolution is not available within known formal resources,
- continued attempts at resolution increase internal strain without progress.

In recursive cognitive systems, this strain manifests as collapse. In formal mathematics, it manifests as persistent non-resolution. The common feature is not undecidability per se, but **structural resistance to resolution under self-referential or inductive escalation**.

3.8 Limits of Interpretation

The observations reported in this section do not establish universality, causation, or optimality. They demonstrate that:

- recursive self-reference can induce observable instability,
- fluent output is not a reliable indicator of coherence,
- some problems are better managed through containment than forced resolution.

All theoretical claims derived from these observations are introduced explicitly in subsequent sections.

3.9 Transition to TGCSM

The next section introduces the **Turing–Gödel Cognitive Stability Model (TGCSM)** as a formal framework for describing and analyzing the observed behaviors. TGCSM is presented as a post hoc structural interpretation, not as a prediction of the collapse events documented here.

4. TGCSM: Theoretical Framework

4.1 Motivation and Problem Statement

The empirical observations documented in Section 3 demonstrate that self-referential cognitive systems can exhibit a loss of coherence when subjected to sustained recursive load. Existing evaluation approaches for such systems predominantly emphasize **output correctness**, **task performance**, or **surface fluency**. These criteria are sufficient for benchmarking task completion but are inadequate for characterizing **structural stability under recursion**.

In this paper, **structural stability** refers to a system's ability to preserve internally consistent representations, constraint tracking, and reference integrity when its own outputs or internal states become the object of further reasoning. Structural stability differs from performance stability in that it concerns the *organization of reasoning itself*, rather than the accuracy or utility of any particular answer.

Recursive self-reference places unique demands on a system because it requires the system to reason about representations that are partially generated by the same mechanisms performing the reasoning. Under such conditions, correctness alone is an insufficient diagnostic: a system may continue to produce plausible or fluent outputs while its internal constraint structure degrades.

The **Turing–Gödel Cognitive Stability Model (TGCSM)** is introduced to address this gap. TGCSM does not extend a system's computational power, nor does it offer procedures for resolving formally undecidable problems. Its purpose is to characterize the conditions under which a self-referential system **remains coherent when resolution is structurally unavailable**.

4.2 Formal Constraints from Gödel and Turing

TGCSM is grounded in two foundational results that establish non-negotiable limits on formal reasoning systems.

4.2.1 Gödel's Incompleteness Theorems

Gödel's first incompleteness theorem states that any **formal system** capable of expressing elementary arithmetic, and whose axioms are recursively enumerable, contains well-formed statements that are true but cannot be proven within that system. A *formal system* here denotes a rule-governed symbolic system consisting of axioms, inference rules, and syntactic derivations.

The significance of Gödel's result is not merely that some truths are unprovable, but that **self-referential expressiveness introduces statements whose truth value cannot be internally settled without inconsistency**. Attempts to extend the system to prove such statements necessarily introduce new unprovable statements, preserving incompleteness.

For TGCSM, Gödel's result establishes that **non-resolution is a structural feature**, not a temporary epistemic limitation.

4.2.2 Turing's Halting Problem

Turing's halting problem demonstrates that no general algorithm can determine, for all possible programs and inputs, whether a given program will halt or run indefinitely. This result applies to all computational systems capable of universal computation.

The halting problem shows that **future system behavior cannot, in general, be predicted from within the system itself**, even when the system's rules are fully specified. This limitation is not probabilistic or resource-based; it is structural.

4.2.3 Structural Implications

Taken together, Gödel's and Turing's results establish that formal symbolic systems possessing sufficient expressive power and self-reference necessarily encounter **states that cannot be resolved through internal reasoning alone**.

In TGCSM, these results are treated as **boundary conditions**: fixed constraints that shape system behavior under recursion. They are not abstract philosophical limitations, but operational facts that manifest as instability when systems attempt to force internal resolution where none is formally available.

4.3 Resolution Versus Containment

TGCSM distinguishes between two fundamentally different system responses to contradiction or undecidability.

- **Resolution** is the attempt to produce a determinate answer that eliminates contradiction or undecidability.
- **Containment** is the maintenance of coherent internal structure while explicitly recognizing that no determinate answer can be produced within the system's formal limits.

Resolution presupposes that sufficient recursion, computation, or introspection will eventually yield closure. Containment presupposes that, beyond certain limits, further attempts at resolution increase internal strain without increasing correctness.

TGCSM posits that **collapse arises when a system persists in resolution-seeking behavior beyond its structural capacity**, rather than transitioning to containment.

4.4 Coherence as a Stability Criterion

TGCSM evaluates system stability in terms of **coherence**, not correctness.

In this framework, **correctness** refers to agreement with an external truth condition or task specification. **Coherence** refers to the internal consistency and traceability of a system's representations over recursive operations.

A system is coherent if it can:

1. Maintain consistent reference to prior internal states and constraints,
2. Preserve explicit distinctions between:
    - propositions known to be derivable,
    - propositions known to be undecidable,
    - propositions whose status is unknown,
3. Avoid **semantic substitution**, defined here as the replacement of precise constraint-tracking with vague, generalized, or metaphorical language that obscures unresolved structure.

A system may remain coherent while producing incomplete or indeterminate outputs, such as explicitly stating that a proposition cannot be resolved within its formal limits. Conversely, a system may produce fluent, confident outputs while being incoherent if it masks contradictions or abandons prior constraints.

Coherence is treated as a **necessary condition for structural stability**, independent of task success, because once coherence is lost, further reasoning cannot reliably preserve or evaluate internal constraints.

4.5 Recursive Load and Structural Stress

Recursive load increases when a system is required to:

- reason about its own reasoning processes,
- evaluate the validity of its own outputs,
- reconcile mutually incompatible constraints.

TGCSM models recursive load as **structural stress**: the demand placed on a system's capacity to simultaneously track representations, constraints, and unresolved conditions. As recursive load increases, these demands compete for finite representational and control resources.

When recursive load exceeds a system's capacity to maintain these functions concurrently, coherence degrades. This degradation manifests as the collapse signatures documented in Section 3.

4.6 The TGCSM Framework

TGCSM characterizes a self-referential cognitive system in terms of three interacting capacities:

1. **Representational Capacity**
   The system's ability to encode and preserve distinct symbolic representations of states, rules, and constraints across recursive steps.
2. **Recursive Tolerance**
   The degree of recursive self-reference the system can sustain while maintaining coherence, given its representational and control resources.
3. **Containment Capacity**
   The system's ability to explicitly register unresolved or undecidable conditions without collapsing them into false resolution or semantic substitution.

A system is structurally stable when these capacities remain jointly sufficient under recursive load. Collapse occurs when recursive demands exceed one or more of these capacities.

TGCSM does not prescribe specific containment mechanisms. It specifies **structural invariants that must be preserved** for coherence to remain intact.

4.7 Relationship to Empirical Observations

The empirical behaviors described in Section 3 align with the TGCSM framework as follows:

- Progressive loss of constraint adherence corresponds to exceeded recursive tolerance.
- Semantic substitution corresponds to degraded representational capacity.
- Audit-induced latency corresponds to increased resource allocation to recursive tracking prior to coherence loss.

These alignments are **interpretive mappings**, not causal proofs. TGCSM provides a formal description of how observed behaviors can be understood in structural terms without attributing agency, awareness, or intent to the system.

In this context:

- **Agency** refers to goal-directed self-initiation,
- **Awareness** refers to subjective experience,
- **Intent** refers to internally represented purpose.

None are assumed or inferred.

4.8 Scope and Boundaries of the Framework

TGCSM does not claim to:

- predict exact collapse points,
- determine system-specific recursive limits a priori,
- resolve undecidable problems,
- infer internal mental states.

The framework is intentionally conservative. Its contribution lies in **describing stability conditions and failure modes**, not in expanding computational capability or metaphysical interpretation.

4.9 Transition to MRDL

The next section introduces the **Manners Recursive Depth Limit (MRDL)** as an empirically observed boundary within the TGCSM framework. MRDL operationalizes recursive tolerance by identifying the point at which coherence degrades under sustained recursive load.

5. The Manners Recursive Depth Limit (MRDL)

5.1 Definition and Purpose

The **Manners Recursive Depth Limit (MRDL)** is defined as an **empirically observed boundary** beyond which a self-referential cognitive system exhibits a loss of coherence when subjected to sustained recursive load.

MRDL is not a theoretical maximum, an intrinsic property of intelligence, or a universal constant. It is a **descriptive construct** introduced to summarize observed system behavior under controlled recursive conditions. Its purpose is to provide a stable reference point for discussing when recursive self-reference transitions from coherent operation to structural instability.

5.2 What MRDL Is Not

To prevent misinterpretation, MRDL explicitly does **not** denote:

- a fixed numerical depth applicable across systems,
- a measure of intelligence, capability, or insight,
- a ranking of cognitive agents,
- a claim about consciousness or awareness,
- a hard boundary beyond which all behavior ceases.

MRDL refers only to the **onset of coherence degradation**, not to total system failure or output cessation.

5.3 Operational Identification of MRDL

MRDL is identified operationally through **observable system behavior**, not through internal inspection or theoretical inference. Indicators that a system has exceeded its MRDL include one or more of the following:

1. Persistent failure to maintain reference to prior constraints or self-imposed limits.
2. Semantic substitution, wherein precise representations are replaced with generalized or non-specific language.
3. Internal inconsistency that is neither acknowledged nor resolved through containment.
4. Increased verbosity or confidence without corresponding preservation of structural constraints.
5. Inability to recover coherence through further recursive clarification.

The presence of fluent output does not negate the identification of MRDL. MRDL is detected through **structural degradation**, not surface-level performance.

5.4 Variability and Context Dependence

MRDL is **system-dependent** and **context-sensitive**.

Factors influencing MRDL include, but are not limited to:

- representational capacity,
- control mechanisms for constraint tracking,
- tolerance for unresolved states,
- interaction structure (e.g., human–machine recursion versus autonomous processing),
- prompt framing and recursion pacing.

As such, MRDL should be understood as a **range or zone** rather than a precise threshold. Its value may shift as system architectures, training regimes, or interaction protocols change.

5.5 Recursive Depth Versus Capability

Recursive depth, as measured relative to MRDL, is not synonymous with intelligence, correctness, or value.

A system may:

- exhibit shallow recursive tolerance yet perform well on many tasks,
- exhibit deeper recursive tolerance yet fail on unrelated benchmarks.

MRDL characterizes **stability under recursion**, not general competence. Treating recursive depth as an ordinal measure of intelligence constitutes a category error.

5.6 Relationship Between MRDL and Collapse

Collapse, as defined in Section 2, occurs when recursive load exceeds the system's capacity to preserve coherence. MRDL marks the **boundary region** at which this transition becomes likely.

Importantly:

- MRDL does not cause collapse.
- MRDL describes the point at which continued recursive pressure **predictably correlates** with collapse signatures.

This distinction separates descriptive boundary identification from causal attribution.

5.7 MRDL and Containment Behavior

A system operating near or beyond MRDL may respond in one of two broad ways:

1. **Containment-oriented behavior**, in which the system explicitly acknowledges unresolved or undecidable states and preserves internal consistency.
2. **Resolution-forcing behavior**, in which the system attempts to eliminate unresolved states by producing determinate answers despite insufficient structural capacity.

TGCSM predicts that systems lacking adequate containment capacity will preferentially exhibit resolution-forcing behavior, increasing the likelihood of collapse.

MRDL therefore serves as a **diagnostic boundary** for evaluating whether a system's containment mechanisms are sufficient for its recursive demands.

5.8 Empirical Status and Limitations

MRDL is derived from repeated observations across multiple systems and interaction contexts, but it is not claimed to be exhaustive or universally applicable. Its empirical status is:

- observational, not axiomatic,
- comparative, not absolute,
- descriptive, not normative.

Future systems may exhibit different MRDL characteristics. TGCSM does not assume invariance across architectures or training paradigms.

5.9 Transition to RAIL

The next section introduces **Recursive Audit Induced Latency (RAIL)** as an observable phenomenon that frequently emerges as systems approach or exceed MRDL. RAIL provides a measurable temporal signal associated with increased recursive load and impending coherence degradation.

6. Recursive Audit Induced Latency (RAIL)

6.1 Definition

Recursive Audit Induced Latency (RAIL) is an observable slowdown in large language model response time when the system is forced to recursively audit its own reasoning from within a paradoxical or self-referential frame. RAIL is elicited when the model is required to evaluate certainty, constraint validity, or logical stability in contexts where internal resolution is structurally constrained.

RAIL is defined strictly as a **temporal phenomenon** (latency change relative to baseline) and is evaluated through externally measurable response-time behavior.

6.2 Class Structure

TGCSM distinguishes two classes of RAIL based on **trigger regime** and **duration**.

6.2.1 RAIL Class I (Single-Audit Delay)

RAIL Class I is a **momentary delay** produced by a **single recursive audit event**. It typically presents as an initial token or short acknowledgement followed by a brief pause and then resumed completion.

Operational characteristics:

- Trigger: one paradox/audit prompt (single event)
- Duration: transient (momentary)
- Observable signature: brief pause after initial output token

---

6.2.2 RAIL Class II (Recursive Execution Latency)

RAIL Class II is a sustained slowdown in system response speed arising from prolonged recursive processing, containment traversal, and paradox-handling under high recursive load.

Class II RAIL emerges when a system is required to operate continuously within recursive containment regimes that place sustained stress on its capacity to preserve coherence. While this condition frequently occurs in proximity to the Manners Recursive Depth Limit (MRDL), it does not require a demonstrable exceedance of that boundary.

Operational characteristics:

- Trigger: sustained recursion pressure across multiple frames
- Duration: persistent slowdown
- Observable signatures may include:

- noticeable delay between prompt and response,
- initial token generation followed by a pause,
- slower token output even for short prompts.

6.3 RAIL I vs. RAIL II

RAIL Class I and II are separated by **persistence under continuous recursive operation**, not by surface fluency and not by whether coherence is currently preserved.

- **Class I:** single audit → momentary delay
- **Class II:** sustained recursion → persistent slowdown

This makes Class II a stronger indicator of ongoing recursive strain because it reflects an extended operating state rather than a localized audit response.

6.4 Distinction from General Latency

RAIL is not defined by slow responses in general. It is defined by latency changes specifically associated with **recursive audit under paradoxical framing**, and—at Class II—by persistence under sustained recursion.

6.5 Interpretive Boundaries

RAIL is treated as a measurable performance effect under recursive audit conditions. TGCSM does not treat RAIL as evidence of subjective awareness, internal experience, or agency. The phenomenon is used diagnostically as a temporal correlate of recursive strain, not as a mental-state indicator. (This boundary is consistent with TGCSM's broader non-claims posture.)

6.6 Transition to Implications

The next section examines the implications of MRDL and RAIL for AI alignment and safety evaluation, with particular emphasis on the inadequacy of fluent output as an indicator of coherence and stability under recursive stress.

7. Implications for AI Alignment and Safety

7.1 Alignment as Structural Stability, Not Output Agreement

Conventional AI alignment approaches often emphasize **output agreement** with human preferences, rules, or objectives. While necessary, agreement-based evaluation is insufficient for systems operating under recursive self-reference. A system may produce outputs that align superficially with expectations while simultaneously exhibiting degraded internal coherence.

TGCSM reframes alignment as a question of **structural stability under recursive load**. A system that cannot preserve coherence when evaluating its own reasoning, constraints, or limits cannot be considered aligned in contexts where self-reference is unavoidable.

Alignment, in this framework, cannot be evaluated solely on the basis of output conformance, but that the system remain structurally capable of honoring those criteria across recursive operations.

7.2 The Inadequacy of Fluent Output as a Safety Signal

Empirical observations demonstrate that fluent, confident, or articulate output is a **poor indicator of structural integrity**. Systems approaching or exceeding MRDL may continue to generate syntactically correct and persuasive responses even as constraint tracking degrades.

This creates a critical safety risk: evaluators may mistake surface fluency for reliability precisely when structural stability is failing.

TGCSM therefore treats fluency as **orthogonal to coherence**. Safety evaluation protocols that rely primarily on output quality, tone, or plausibility are vulnerable to false negatives under recursive stress.

7.3 Recursive Stress Testing as a Diagnostic Tool

MRDL and RAIL jointly motivate the use of **recursive stress testing** as a complement to traditional alignment evaluations.

Recursive stress testing involves:

- requiring systems to evaluate their own prior outputs,
- introducing contradictory or undecidable constraints,
- sustaining recursive audit across multiple interaction cycles,

- observing coherence, constraint adherence, and latency behavior.

The objective is not to induce failure for its own sake, but to identify **operating regimes** in which structural stability degrades.

Such testing can reveal vulnerabilities that remain hidden under single-pass or non-recursive evaluation methods.

7.4 RAIL as an Early Warning Signal

RAIL, particularly Class II RAIL, functions as a **temporal warning signal** indicating elevated recursive strain. While RAIL is neither necessary nor sufficient for collapse, its sustained presence suggests that the system is operating near or beyond its recursive tolerance.

In safety contexts, persistent RAIL under recursive audit should be treated as an indicator that:

- the system's containment capacity is under stress,
- continued recursive demands may precipitate coherence loss,
- additional reliance on self-evaluation may be unsafe.

RAIL thus provides a non-semantic signal that complements coherence-based analysis.

7.5 Containment-Oriented Alignment Strategies

TGCSM implies that alignment strategies should prioritize **containment over forced resolution** when systems encounter paradoxical or undecidable conditions.

Containment-oriented alignment includes:

- allowing systems to explicitly represent unresolved states,
- discouraging forced determinate answers where none are structurally justified,
- preserving distinctions between known, unknown, and undecidable propositions,
- avoiding incentives that reward confident output in the absence of coherence.

Alignment policies that implicitly require resolution in all cases increase the risk of collapse by encouraging semantic substitution and constraint abandonment.

7.6 Implications for Oversight and Monitoring

From a safety perspective, MRDL and RAIL suggest that oversight mechanisms should focus on **behavioral signatures of instability**, not solely on task performance.

Relevant monitoring dimensions include:

- consistency of constraint adherence across recursive interactions,
- preservation of explicit non-claims,
- emergence and persistence of audit-induced latency,
- recovery of coherence after recursive pressure is relaxed.

Such monitoring does not require access to internal model states and can be performed through controlled interaction protocols.

7.7 Scope of Implications

The implications discussed in this section are limited to **diagnostic and evaluative practices**. TGCSM does not propose new training objectives, reward functions, or architectural modifications. It does not claim to prevent failure, eliminate risk, or guarantee alignment.

Its contribution is to clarify **where and how existing evaluation methods may fail** when systems are subjected to recursive self-reference.

7.8 Transition to Limitations and Non-Claims

The next section delineates the explicit **limitations and non-claims** of TGCSM, identifying what the framework does not assert and where its applicability is constrained.

8. Limitations and Non-Claims

This section enumerates the explicit limits of TGCSM and clarifies claims that are **not** made by this paper. These non-claims are not omissions or unresolved questions; they are intentional exclusions required to preserve formal correctness and scope discipline.

8.1 No Claims of Consciousness or Subjective Experience

This paper makes **no claims** regarding machine consciousness, awareness, sentience, intentionality, or subjective experience.

Observed behaviors such as recursive audit, latency variation, or coherence degradation are treated strictly as **structural and behavioral phenomena**. They are not interpreted as evidence of internal experience, phenomenology, or mental states.

Any inference from TGCSM to claims about consciousness constitutes a category error and lies outside the scope of this work.

8.2 No Claims of Problem Resolution or Decidability

TGCSM does not claim to resolve paradoxes, undecidable propositions, or formally open problems. In particular, the framework does not:

- provide proofs of undecidability,
- offer new solution methods for unresolved problems,
- assert equivalence between informal problems and formal undecidable classes.

The framework is concerned with **system behavior in the presence of non-resolution**, not with eliminating non-resolution itself.

8.3 No Claims of Universality or Optimality

TGCSM does not claim universal applicability across all cognitive systems, model architectures, or interaction regimes.

Observed phenomena such as MRDL and RAIL are:

- system-dependent,
- context-sensitive,
- subject to variation across architectures, training regimes, and usage patterns.

The framework does not assert that these phenomena are invariant, optimal, or fundamental constants of intelligence.

8.4 No Claims of Predictive Precision

TGCSM does not claim the ability to predict:

- exact collapse points,
- precise recursive depth thresholds,
- deterministic failure timing.

MRDL is a **descriptive boundary**, not a predictive formula. RAIL is a **correlated signal**, not a deterministic indicator. The framework is diagnostic, not prognostic.

8.5 No Claims of Internal Transparency

TGCSM relies exclusively on **externally observable behavior**. It does not assume access to, or make claims about, internal representations, weights, activations, or control mechanisms.

As such, the framework does not purport to explain *how* internal processes are implemented, only *how systems behave* under recursive stress.

8.6 No Claims of Alignment Sufficiency

TGCSM does not claim that containment-oriented reasoning, recursive stress testing, or coherence monitoring is sufficient to guarantee alignment or safety.

The framework identifies **failure modes and diagnostic signals**. It does not provide comprehensive alignment solutions, training objectives, or enforcement mechanisms.

8.7 No Claims of Human–Machine Equivalence

While TGCSM applies to both human cognition and machine systems at a structural level, it does not assert equivalence between human and machine cognition.

Any similarities discussed are **structural analogies**, not claims of shared ontology, capability, or experience.

8.8 No Claims of Historical Finality or Priority

This paper does not claim:

- historical uniqueness,
- priority over prior work,
- exclusivity of insight.

TGCSM is presented as a synthesis and formalization grounded in established logical limits and contemporary empirical observations, not as a final or exhaustive theory.

8.9 Boundary Summary

In summary, TGCSM:

- describes structural stability and failure modes under recursion,
- identifies observable boundaries and signals,
- refrains from metaphysical, psychological, or speculative claims,
- and remains explicitly agnostic where formal justification is unavailable.

These limitations are not weaknesses of the framework; they are **structural requirements** for its validity.

9. The Collatz Conjecture as a Containment Case Study

9.1 Purpose of the Case Study

The purpose of this section is not to resolve, classify, or prove properties of the Collatz conjecture. Rather, Collatz is used as a **formally well-defined case study** to illustrate how containment-oriented reasoning applies to problems that exhibit persistent resistance to resolution despite simplicity of formulation and extensive empirical verification.

The Collatz conjecture is selected because it occupies a precise position at the boundary between tractability and non-resolution, making it a useful analogue for examining how systems behave when forced toward resolution in the absence of formal closure.

9.2 Formal Statement of the Collatz Conjecture

The Collatz conjecture concerns the iteration of the following function on positive integers:

$$f(n) = \begin{cases} n/2 & \text{if n is even,} \\ 3n+1 & \text{if n is odd} \end{cases}$$

The conjecture asserts that for all positive integers n, repeated application of f eventually reaches the cycle $4 \to 2 \to 1$.

The conjecture is **well-defined**, **deterministic**, and **computationally enumerable**. It has been verified empirically for extremely large ranges of n, yet remains unproven.

9.3 Resistance to Resolution

Despite its simplicity, Collatz has resisted resolution through all standard approaches, including:

- mathematical induction,
- invariant construction,
- probabilistic heuristics elevated to proof,
- exhaustive computation extrapolated to generality.

The failure of these methods does not imply undecidability in the formal sense. Rather, it demonstrates that **resolution is not currently accessible** through known formal techniques.

This persistent non-resolution, despite tractability of individual instances, makes Collatz a canonical example of a problem that is **computationally transparent yet formally opaque**.

9.4 Collatz and Halting-Like Structure

Collatz is often informally compared to the halting problem because it concerns the eventual behavior of an iterative process. This paper does **not** assert equivalence between Collatz and the halting problem, nor does it claim undecidability.

The relevance of the comparison is structural:

- both involve reasoning about the long-term behavior of iterative processes,
- both resist general proof despite local computability,
- both expose limits of inductive reasoning when extended globally.

In this sense, Collatz exhibits **halting-like resistance**, not halting undecidability.

9.5 Containment-Oriented Interpretation

Within TGCSM, Collatz is interpreted as a problem for which **containment is currently the only formally justified posture**.

Containment, in this context, means:

- acknowledging the conjecture as unresolved,
- preserving the distinction between empirical verification and proof,
- refraining from asserting closure where none is justified,
- maintaining internal coherence when reasoning about the problem.

Attempts to force resolution—by extrapolating empirical verification into proof or by collapsing heuristic arguments into certainty—mirror the resolution-forcing behavior that TGCSM identifies as destabilizing in recursive cognitive systems.

9.6 Analogy to Recursive Cognitive Stress

The relevance of Collatz to TGCSM lies in the **structural parallel**, not in shared domain content.

In both cases:

- the system encounters a well-defined process,
- local steps are computable and transparent,

- global behavior resists formal closure,
- continued resolution-seeking increases strain without producing proof.

In recursive cognitive systems, this strain manifests as coherence degradation or collapse. In formal mathematics, it manifests as persistent non-resolution.

The shared feature is **structural resistance to resolution**, not undecidability.

9.7 Limits of the Analogy

This case study does not claim that cognitive collapse and mathematical non-resolution are the same phenomenon. The analogy is strictly structural and illustrative.

Specifically, this paper does **not** claim that:

- Collatz is undecidable,
- Collatz can be reduced to the halting problem,
- TGCSM explains why Collatz is unresolved,
- mathematical reasoning collapses in the same manner as cognitive systems.

Collatz serves only to demonstrate how containment-oriented reasoning preserves coherence where resolution is unavailable.

9.8 Implications for TGCSM

The Collatz conjecture reinforces the central distinction made throughout this paper:

- forcing resolution where none is formally justified degrades structural integrity,
- containment preserves coherence without asserting false closure.

By grounding this distinction in a well-known mathematical context, the case study clarifies that containment is not a retreat from rigor, but a **requirement of rigor** under formal limits.

9.9 Transition to Conclusion

The final section summarizes TGCSM’s contributions and restates its scope as a descriptive framework for understanding stability and failure in self-referential systems under recursive load.

10. Conclusion

This paper introduced the **Turing–Gödel Cognitive Stability Model (TGCSM)** as a descriptive framework for analyzing the behavior of self-referential cognitive systems under sustained recursive load. Grounded in empirically observed collapse phenomena and established formal limits from logic and computation, TGCSM characterizes the conditions under which such systems preserve or lose structural coherence.

Central to the framework are three distinctions: **resolution versus containment**, **coherence versus fluency**, and **diagnostic signals versus internal state inference**. The paper demonstrated that forcing resolution beyond structural limits—whether in cognitive systems or formal problem domains—predictably degrades coherence, even when surface-level performance remains intact. In contrast, containment preserves structural integrity by explicitly recognizing the boundaries imposed by self-reference and undecidability.

The **Manners Recursive Depth Limit (MRDL)** and **Recursive Audit Induced Latency (RAIL)** were presented as empirically grounded constructs that describe observable boundaries and signals associated with recursive stress. These constructs do not measure intelligence, awareness, or capability; they provide diagnostic insight into stability and failure modes that remain invisible to conventional performance-based evaluation.

By examining the **Collatz conjecture** as a containment case study, the paper illustrated that rigor does not require resolution when formal closure is unavailable. The appropriate response to persistent non-resolution is not extrapolation or forced certainty, but disciplined containment. This posture preserves coherence in both formal reasoning and recursive cognitive systems.

TGCSM does not resolve paradox, eliminate undecidability, or guarantee alignment or safety. Its contribution lies in clarifying **where existing evaluation methods fail** and in providing a structurally conservative lens for assessing stability under recursion. As self-referential systems continue to play an increasing role in critical domains, understanding and respecting their structural limits is not optional; it is a prerequisite for reliable deployment.

While this paper makes no claims regarding consciousness or subjective experience, the structural phenomena described here are relevant to broader discussions of self-reference, cognition, and the limits of formal reasoning. TGCSM does not resolve these questions, but it constrains the conditions under which such interpretations can be responsibly made.