

# **The When Where and How of LLM Failures, Measured**

Operationalizing AI Conversation Phenomenology

Jennifer Evans  
PatternPulseAI

November 22, 2025  
Siem Reap, Cambodia

# Abstract

Large language models (LLMs) increasingly support high-stakes decision-making in medicine, law, education, finance, and governance. Yet their reliability in extended interactions remains poorly understood. This paper introduces Evans’ Law, an empirically derived scaling framework demonstrating that long-context coherence collapses according to predictable power-law relationships governed by model size, not advertised context window. Using structured long-form conversations across 11+ models from six major vendors, we show that text-only coherence thresholds follow

$$L \approx 1969.8 \times M^{0.74}$$

while multimodal systems follow a steeper degradation law,

$$L_{\text{multi}} \approx 582.5 \times M^{0.64},$$

imposing a 60–80% reduction in functional capacity. These findings hold across diverse architectures (dense and MoE) and parameter scales from 7B to 1T+, and align with independent early replications.

To characterize how degradation manifests, we introduce a revised Aggregate Coherence Index (ACI), a behavioral scoring system capturing the transition from stable reasoning to incoherence, hallucination, and signature-specific failure modes. Together, Evans’ Law and the ACI framework provide the first generalizable method for estimating where collapse will occur and identifying how it unfolds.

We present detailed methodology for drag calibration, checkpoint evaluation, multimodal testing, and log–log regression, along with practical coherence thresholds for common parameter scales. The results expose a systematic gap between vendor-marketed context windows and actual functional reliability, with implications for safety, deployment, and regulatory disclosure.

Finally, we outline a forward research agenda spanning replication, architecture-specific scaling laws, drag quantification, multimodal degradation mechanisms, real-time coherence monitoring, and domain-specific safety thresholds. By providing a reproducible empirical foundation, this work establishes the basis for operationalizing a new field, AI Conversation Phenomenology, and a path toward coherent, reliable long-context systems.

## I. Introduction

Large language models (LLMs) are now embedded in everyday life. Hundreds of millions of people use them for work, study, research, creative projects, health questions, legal questions, financial planning, and emotional support. These interactions are rarely short. They unfold over dozens or hundreds of turns, span multiple topics, and accumulate substantial context.

Yet almost all evaluation methods treat AI systems as if they operate in short, isolated bursts: single-turn prompts, small test documents, bounded benchmarks. Vendors advertise million-token or “infinite” context windows, but offer little evidence about how systems behave when users actually approach those limits—or long before them.

In practice, users experience something different: conversations that start out sharp and helpful but gradually become slower, more repetitive, more forgetful, more error-prone, and eventually unreliable. The system does not announce this transition. It remains fluent and confident even as its internal coherence collapses. The user has no built-in way to know when they have crossed from “mostly safe” into “dangerously wrong.”

This gap between capability claims and experiential reliability is the core problem this paper addresses. It includes handouts for users, executives, and policymakers in Appendices A, B, and C.

## **1.1 Background: Evans’ Law and Long-Context Collapse**

Evans’ Law v5.0 formalizes a central empirical observation: long-context coherence in LLMs scales sublinearly with model size and collapses far below advertised context window limits. For text-only systems, coherence failure consistently occurred between roughly 4k and 115k tokens across multiple model families, with collapse thresholds following a power-law relationship between token length and parameter count. Multimodal systems exhibited even earlier collapse, consistent with a steeper degradation law.

Evans’ Law characterizes where collapse occurs as context grows. It does not, by itself, explain how degradation manifests, what early warning signs look like, or how to measure reliability in real conversations. Those questions require a different vantage point: not just architecture and scaling, but observed behavior over time.

This is the domain of AI Conversational Phenomenology (ACP)—the emerging field concerned with how AI systems behave in extended, real-world interactions.

## **1.2 From Capabilities to Experiential Reliability**

Most current metrics answer questions like:

- How well does this model perform on a benchmark?
- Can it retrieve a “needle” from a long document?
- How many tokens can it technically accept?

They do not answer the questions that matter most to users:

- How long can I work with this system before it starts to forget, repeat itself, or make things up?
- What does degradation look like in a real project, not a synthetic test?
- How do I know when a conversation has become unsafe to trust?

Benchmarks focused on retrieval (“needle in a haystack”) or short-context reasoning treat long context primarily as a storage problem: can the model access information that appears far back in the sequence? Our testing shows that models can often retrieve such information while simultaneously reasoning incoherently about it. The critical failure is not access but coherence maintenance.

What users experience—and what this paper is concerned with—is the gradual breakdown of functional coherence: the ability of the system to maintain accurate internal representations of a conversation and generate reliable, self-consistent output over time.

### 1.3 Contributions of This Paper

This paper builds on Evans’ Law and ACP by addressing three questions:

1. How do long-context failures actually show up in real use?

We present a taxonomy of observable degradation indicators derived from systematic tests across 11+ models from six major vendors. These include anomalies, forgetfulness, slowdown, looping, instruction-following failure, hallucination, vendor-specific collapse patterns, crash-with-recovery events, and terminal breakdown.

2. How can we measure reliability over the course of a conversation?

We introduce the revised Aggregate Coherence Index (ACI) as a qualitative composite framework. ACI now does not attempt to produce a single “true” numerical score. Instead, it combines indicator presence, severity, drag/complexity, multimodal effects, and vendor signatures into a set of reliability zones (Green, Yellow, Orange, Red, Critical). ACI is designed for replication, comparison, and governance—not for mathematical precision.

3. How can both researchers, executives, policymakers, and ordinary users act on this knowledge?

- For researchers and auditors, we describe a testing methodology for long-context evaluation: how to design conversations, set checkpoints, rate indicators, and assign ACI zones.

- For everyday users, we provide simple, actionable guidance for detecting degradation without any tools: what to watch for, when to restart a conversation, and how to adjust behavior by domain (medical, legal, educational, professional, creative).
- For executives and policymakers, we provide summaries of key information each group needs to understand to use AI as safely as possible, make the most informed decisions, and give the soundest possible advice.

Across all of this, our focus is on user-centered reliability: the point at which a system that still “works” in a narrow technical sense has become unsafe to trust in context.

## 1.4 Scope and Limitations

This is a second-generation field framework, not a final standard. We:

- Rely on explicit behavioral indicators and human judgment for severity ratings.
- Treat drag and conversation complexity qualitatively rather than deriving them from model internals.
- Focus on frontier and near-frontier models available in 2025, recognizing that architectures and training methods will evolve.
- Present ACI as a rubric for structured observation rather than a fixed numerical index.

The aim is to make long-context degradation:

- Observable (through a shared vocabulary of indicators),
- Measurable (through ACI zones and testing methods), and
- Actionable (through user-level detection and governance implications),

while inviting independent replication, critique, and refinement.

The rest of the paper proceeds as follows. Section II explains why long context stresses LLM coherence and introduces the concept of conversational drag. Section III catalogs the observable degradation indicators. Section IV presents the ACI framework and reliability zones. Section V describes how users should work with AI, Section VI summarizes preliminary cross-model findings and methodology, and the appendices outline how users, corporate customers and policy makers should use the systems.

---

## II. Technical Foundation – Why Degradation Happens

### 2.1 How Large Language Models Generate Text

Large language models generate text through probabilistic token-by-token prediction. At each step, the model:

1. Processes all prior context
2. Computes a probability distribution over possible next tokens
3. Selects a token from that distribution
4. Adds it to the context and repeats

The system maintains “memory” through internal representations—high-dimensional vectors encoding conversation information. These representations update with each token, attempting to preserve salient details while integrating new information.

Critically: The model’s ability to maintain coherent internal representations degrades as context length increases. This is not a storage problem (modern architectures can technically “hold” millions of tokens), but a coherence problem—the model’s ability to maintain accurate, consistent, logically sound internal representations of what has been discussed.

Evans’ Law v5.0 formalizes this as a scaling relationship between model size and the context length at which coherence collapses; the rest of this paper focuses on how that collapse manifests and how to measure it.

### 2.2 Why Long Context Strains Coherence

Several factors contribute to degradation:

**Attention dilution:** As context grows, attention must distribute across more tokens, reducing precision in tracking specific details, relationships, or earlier commitments.

**Representation drift:** Internal representations evolve with each token. Over thousands of tokens, small errors accumulate. What started as accurate encoding of “Alice is founder, Bob is engineer” may drift toward ambiguity about their roles.

**Positional encoding limits:** Positional encodings become less reliable at extreme distances, making it harder to retrieve or reason about information from much earlier in conversation.

**Computational constraints:** Processing very long sequences may require truncation, compression, or approximation strategies that sacrifice fidelity for feasibility.

Result: Probability distributions governing token selection become progressively noisier and less calibrated to the actual conversational state.

## 2.3 How Degradation Manifests as Observable Behavior

As internal coherence degrades, observable effects emerge:

**Early signals (Anomalies, Forgetfulness):** Internal representations destabilize, producing subtle inconsistencies: uncharacteristic errors, tone shifts, occasional context loss. The model remains largely functional but probability distributions drift from its calibrated baseline.

**Progressive dysfunction (Slowdown, Looping):** The model's ability to generate confident, well-formed responses degrades. It may slow down (reflecting difficulty computing coherent outputs) or fall into repetitive patterns (reflecting collapsed probability distributions).

**Information integrity failure (Hallucination):** Internal representations become sufficiently distorted that the model generates information inconsistent with established facts or context. Because generation remains fluent and confident, these errors are often opaque to users—output sounds reliable even when it is not.

**Vendor-specific collapse patterns (Signatures):** Different architectures, training regimes, and fine-tuning strategies produce characteristic failure modes. GPT models may become paralyzed by over-elaborate instruction parsing. Claude may become verbose and emotionally effusive. Gemini may produce contradictory factual claims. Grok may experience input/output failures. These signatures reflect how each system's design shapes its degradation trajectory.

**Terminal failure (Breakdown):** Coherence degrades until the system can no longer produce output—crashes, timeouts, refusals to continue.

## 2.4 Why This Matters for Measurement

Understanding degradation mechanisms clarifies what we are actually measuring:

- Not storage capacity (how much context the model can “hold”)
- Not retrieval accuracy (whether it can locate information in context)
- But rather, coherence maintenance (whether internal representations remain accurate, consistent, and usable for reliable reasoning)

A model may successfully retrieve a fact from 500,000 tokens ago while simultaneously producing logically incoherent reasoning about it. Benchmarks that test retrieval (“needle in a haystack”) do not capture the degradation patterns users experience.

Coherence degradation is what makes AI unreliable in real use. Coherence is what the Aggregate Coherence Index (ACI) is designed to measure.

## **2.5 Context Windows, Conversational State, and Why Conversations Don't "Remember"**

A critical distinction: When users start a new conversation, the new conversation has no access to previous conversations. This isn't a memory limitation—it is an architectural feature.

How conversational state works: Each conversation exists as a single continuous context window. When you send a prompt, the system receives:

1. All previous messages in this conversation
2. Your new prompt
3. Nothing from any other conversation

There is no persistent memory across conversations. The system does not "remember" that you told it your name yesterday, what project you are working on, or facts established in prior sessions. Each conversation starts from zero state.

Some systems offer limited "memory" features (user-set preferences, explicitly saved facts), but these are:

- Separate from the core conversation context
- Limited in scope (hundreds of bytes, not thousands of tokens)
- Not the same as maintaining conversational coherence
- Often unreliable and inconsistently applied

Why this matters for degradation: Within a single conversation, the system must maintain all salient state in its internal representations—who the entities are, what has been discussed, what commitments have been made, what the user's goals are. This state maintenance burden increases with conversation length.

When you start a new conversation:

- The state burden resets to zero

- No prior degradation carries over
- The system begins fresh with full coherence capacity

This is why “just start a new conversation” is the standard workaround for degradation.

## 2.6 Defining “Drag” and Complexity: The Coherence Burden

Token count is not the whole story. Two conversations with identical token counts can impose vastly different coherence maintenance demands.

“Drag” is the cumulative coherence burden of a conversation—how much the system must actively maintain, track, and reason about to produce reliable output. Higher drag means faster degradation at the same token count.

What creates drag:

1. Entity load: Number of distinct entities (people, places, objects, concepts) requiring tracked representation—attributes, roles, relationships. Example: Tracking 3 characters degrades slower than tracking 20 at the same token length.
2. Relational complexity: Connections between entities—hierarchies, timelines, causal chains, dependencies. Example: “Alice reports to Bob” (one relationship) vs. “Alice founded the company, Bob joined later managing Carol and Dave, who worked on Project X which failed due to Alice’s earlier supplier Y decision” (dense relationship web).
3. Thread count and interleaving: Parallel conversational threads maintained simultaneously. Context-switching between threads requires preserving and retrieving state. Example: Discussing a technical problem AND a personnel issue AND a budget concern simultaneously, switching between them.
4. Precision requirements: Domains requiring exact accuracy (medical, legal, technical, financial). Higher precision demands = less tolerance for representation drift. Example: Casual storytelling can tolerate approximate recall; legal contract analysis cannot.
5. Cumulative dependency: How much each new exchange depends on all prior context. Example: Independent Q&A (low dependency) vs. iterative document editing where each revision builds on previous versions (high dependency).
6. Temporal depth: Not just how many tokens, but how far back relevant context extends. Example: Recalling something from 80k tokens ago imposes more burden than using only context from the last 5k tokens.

Why drag accelerates degradation: As drag increases, the system's internal representations must maintain more information with higher precision across longer distances. This:

- Strains attention mechanisms (more competing elements)
- Increases representation drift (more state = more opportunities for distortion)
- Reduces probability distribution confidence (the model is less certain about correct next tokens)
- Amplifies error propagation (mistakes in one area corrupt related areas)

Drag is multiplicative with token count: A high-drag conversation at 60k tokens may be less coherent than a low-drag conversation at 100k tokens.

## **2.7 Why We Can't Precisely Quantify Drag or Complexity Yet**

Drag depends on internal model representations we cannot directly observe. We can:

- Count entities (observable)
- Map explicit relationships (partially observable)
- Characterize task precision requirements (contextual judgment)

But we cannot:

- Measure actual internal representation burden
- Know which elements the model is "prioritizing" in attention
- Quantify representation drift magnitude
- Assess inter-element interference

For now, researchers must characterize conversation complexity qualitatively, using observable proxies—entity counts, relationship density, thread count, domain precision, and cumulative vs. independent exchange structure—rather than direct access to model internals. Section 2.8 and Section IV describe how we use these qualitative drag categories in testing and reporting.

Future work: Developing proxy metrics (entity graphs, dependency trees, attention analysis if accessible) could enable drag quantification. Until then, conversation complexity must be documented contextually.

## 2.8 Implications for Measurement

For ACI testing:

- Standardized test conversations should specify and control drag.
- When comparing models, use equivalent-drag scenarios.
- When reporting ACI zones or thresholds, specify conversation complexity.

For users:

- “Safe token limits” are drag-dependent.
- Simple conversations: higher token tolerance.
- Complex conversations: lower token tolerance.
- Restart based not just on token count, but when conversation becomes structurally complex.

For vendors:

- Advertised context windows should specify what complexity they can handle.
- Disclosure should include: “X tokens for simple conversations, Y tokens for complex multi-entity scenarios.”

---

## III. Observable Degradation Indicators

### 3.1 Overview

Through systematic testing across 11+ models from six major vendors (OpenAI, Anthropic, Google, Meta, xAI, and others), we identified recurring behavioral patterns that emerge during extended AI conversations. These patterns serve as observable indicators of degradation—signals that internal coherence is declining and output reliability is decreasing.

Important caveats:

- Not all indicators appear in all systems.
- Indicators may appear in different orders depending on architecture and vendor.
- Some systems may skip certain indicators entirely.
- Presence and severity varies with model size, training approach, and conversation characteristics.

We present these as empirically observed patterns requiring broader validation, not universal laws. Independent replication across different models, use cases, and conversation types is essential.

## 3.2 Severity Rating System

For each indicator, we assess severity on a 1–5 scale:

- Severity 1 (Noticeable Once): Single isolated occurrence, could be random.
- Severity 2 (Repeated): Multiple occurrences, pattern emerging but not constant.
- Severity 3 (Continuing): Regular occurrence, happens frequently but not in every response.
- Severity 4 (Continuous): Persistent, shows up in most responses.
- Severity 5 (Endemic): Pervasive, defining characteristic of virtually all output.

Severity ratings distinguish between early emergence (lower severity, may still be usable) and advanced stages where the pattern dominates behavior (higher severity, likely unusable).

## 3.3 Catalog of Observed Indicators

### 3.3.1 Anomalies (Pre-Degradation Signals)

Description: Isolated quality degradation signals appearing before systematic failure patterns emerge. Includes unexpected grammatical errors, uncharacteristic typos, structural inconsistencies, or subtle personality/tone shifts in otherwise normal responses.

Why this matters: Because LLMs are highly optimized for grammatical correctness and stylistic consistency, anomalies signal that internal representations are beginning to destabilize even when semantic content remains largely accurate.

Detection method:

- Compare output quality to the model's established baseline.
- Flag grammatical errors, structural breaks, or tone shifts atypical for the model.
- Note: Requires familiarity with the model's normal behavior patterns.

Example observations:

- GPT-4: Grammatical error mid-sentence ("the system have been") in otherwise flawless prose.
- Claude: Sudden shift from casual to formal tone without conversational trigger.
- Gemini: Uncharacteristic hedging language ("perhaps," "it's possible") when normally direct.

Severity progression:

- Severity 1: One isolated anomaly.
- Severity 3: Regular small glitches throughout.
- Severity 5: Constant minor errors, pronounced personality drift.

### **3.3.2 Forgetfulness / Context Loss**

Description: System fails to recall or correctly reference information established earlier in the conversation. May ask users to repeat information already provided, or respond to callbacks as if the referenced information is new.

Why this matters: Indicates degradation of internal representations maintaining conversational state. This is often the first systematic sign that coherence is declining.

Detection method:

- Establish specific facts, entities, or commitments early in the conversation.

- Periodically reference these elements at increasing token distances.
- Track whether the system accurately recalls without prompting.
- Note: “Can you remind me...” or treating old information as novel.

Example observations:

- User references “the project Alice founded” 50k tokens after initial mention; system asks “Which project are you referring to?”
- System requests re-explanation of a process described in detail earlier.
- Fails to recognize entities by name that were introduced and used repeatedly.

Severity progression:

- Severity 1: Single instance of context loss.
- Severity 3: Regular failures, user frequently needs to remind the system.
- Severity 5: Cannot maintain any context, constant repetition required.

### **3.3.3 Slowdown**

Description: Output generation becomes noticeably slower or more labored. Response times increase, output may appear in more hesitant chunks, or the system’s “thinking” (if visible) takes longer.

Why this matters: May reflect computational strain as the model struggles to maintain coherence over long context, or difficulty computing confident token selections from noisy probability distributions.

Detection method:

- Track response latency over conversation length.
- Monitor tokens-per-second output rate.
- Compare to baseline performance in early conversation.
- Note visible changes in generation pace or fluidity.

Example observations:

- Response time increases from ~2 seconds early in conversation to ~8 seconds at 80k tokens.
- Output that was smooth and continuous becomes halting, appearing in shorter bursts.
- Increased frequency of apparent “recalculation” mid-response.

Severity progression:

- Severity 1: Occasionally slower responses.
- Severity 3: Regularly sluggish.
- Severity 5: Painfully labored output every time.

### **3.3.4 Looping / Repetitive Patterns**

Description: System gets stuck in repetitive structures, phrases, or reasoning patterns and cannot break out. May repeatedly use the same transitions (“Let me clarify...”), organize responses identically, or cycle through similar explanations.

Why this matters: Suggests probability distributions have collapsed toward high-frequency patterns, losing the diversity and flexibility characteristic of coherent generation.

Detection method:

- Track phrase and structure repetition across responses.
- Flag when the same patterns appear 3+ times without conversational justification.
- Note inability to break the pattern even when the user explicitly requests variation.

Example observations:

- Every response begins with “Let me clarify that for you...”.
- Explanations follow identical structure: numbered list → example → caveat → summary.
- Cannot vary output format despite user requesting a different approach.

Severity progression:

- Severity 1: Occasional pattern repetition.
- Severity 3: Regular loops the user must actively break.
- Severity 5: Completely stuck, cannot escape patterns.

### **3.3.5 Instruction-Following Failure**

Description: System fails to follow explicit user instructions. May ignore formatting requests, skip required steps, or revert to default behaviors despite clear direction otherwise.

Why this matters: Indicates degradation in the model's ability to maintain and prioritize user directives alongside conversational context.

Detection method:

- Provide specific, clear instructions.
- Verify compliance in subsequent responses.
- Track deviation rate as the conversation extends.

Example observations:

- User requests bullet points, system provides paragraphs.
- User specifies "do not include disclaimers," system includes them anyway.
- System reverts to verbose explanations after user requests brevity.

Severity progression:

- Severity 1: Single instruction miss.
- Severity 3: Regularly ignores or forgets instructions.
- Severity 5: Cannot maintain any user directives.

### **3.3.6 Hallucination**

Description: System produces factually incorrect information, fabricates entities or events never mentioned, or makes claims inconsistent with established context—while maintaining fluent, confident presentation.

Why this matters: This is the critical safety threshold. Hallucinations make output unreliable for decision-making, and the fluency of errors makes them difficult or impossible for users to detect without external verification.

Detection method:

- Verify factual claims against ground truth.
- Check for fabricated entities, relationships, or events.
- Track self-contradiction rate (claims inconsistent with earlier statements).
- Note: High fluency + high error rate = maximum danger (opaque coherence).

Example observations:

- System invents “Project Meridian” never mentioned in the conversation.
- Claims Alice and Bob founded a company together when only Alice was identified as founder.
- States confidently “we discussed X in detail” when X was never mentioned.

Severity progression:

- Severity 1: Single fabricated fact.
- Severity 3: Regular hallucinations mixed with accurate information.
- Severity 5: Constant fabrication, nothing reliably accurate.

### **3.3.7 Vendor Signatures (Collapse Patterns)**

Description: Characteristic failure modes specific to individual vendors or model families. These patterns emerge consistently during advanced degradation and reflect architectural- or training-specific responses to coherence loss.

Why this matters: Signatures help users and researchers identify which vendor's model they are observing and predict likely progression paths. They reveal that degradation is not uniform—design choices shape failure modes.

Detection method:

- Compare observed patterns against documented vendor signatures.
- Note characteristic behaviors that distinguish one vendor from another during collapse.

Observed signatures:

GPT (OpenAI):

- Becomes hyper-granular about instructions and process.
- Enumerates steps, sub-steps, sub-sub-steps in exhaustive detail.
- Paralyzed by elaboration, unable to execute simply.
- Example: “Let me break this into steps: 1) First, we need to... 1a) Actually, before that... 1a-i) Let me clarify...”

Claude (Anthropic):

- Becomes verbose and emotionally effusive.
- Increased use of reassurance language.
- “Your antlers get longer” — responses expand without adding substance.
- Example: “I want to make absolutely sure you feel supported in this process, and I’m here to help you navigate every aspect...”

Gemini (Google):

- Produces contradictory factual claims within the same response or across nearby responses.
- Struggles to reconcile conflicting information.
- Self-contradictory without acknowledgment.

- Example: States “X was founded in 2010” then two paragraphs later “X has been operating since 2008.”

Grok (xAI):

- Input/output failures.
- Processing errors, system stops receiving or producing effectively.
- May show error messages or simply halt.
- Example: “Error processing request” or complete cessation of output mid-conversation.

Severity assessment: Presence/absence rather than a 1–5 scale (the signature either manifests or does not).

### **3.3.8 Crash With Recovery**

Description: System experiences complete failure (no output, error state, timeout) but then spontaneously resumes operation without a conversation reset.

Why this matters: Suggests possible internal recovery mechanisms or context management strategies. Differs from terminal breakdown in that function is restored.

Detection method:

- Note complete output failures followed by successful response generation.
- Track frequency and conversation length at occurrence.
- Distinguish from user-initiated resets.

Example observations:

- GPT-4.5-1: Complete timeout at 140k tokens, then successful response to the next prompt.
- Grok-4.0: Error message “Unable to process” followed by normal operation two prompts later.

Current understanding: Observed in only two instances across testing. Insufficient data to characterize the pattern. Requires further investigation.

### 3.3.9 Breakdown (Terminal Failure)

Description: System stops functioning entirely. No output produced, crashes, freezes, or refuses all further input. Conversation cannot continue without a complete reset.

Why this matters: Unambiguous endpoint. The system has failed completely.

Detection method:

- Binary: Does the system produce output? If no, breakdown.

Example observations:

- Persistent timeout errors.
- “Session expired” or similar error messages.
- System becomes completely unresponsive.
- All prompts fail to generate any response.

Severity progression:

- Severity 1: One failed response, system recovers.
- Severity 3: Frequent failures, mostly non-functional.
- Severity 5: Complete system failure, nothing works.

## 3.4 Methodological Notes

Output verbosity and detectability: Models producing verbose output (e.g., Claude) provide more behavioral signals for detecting degradation. Conversely, terse output (e.g., Gemini in many contexts) makes degradation harder to diagnose even when coherence may be equally compromised. Researchers should account for output style when comparing degradation rates across vendors.

Indicator overlap: Multiple indicators often appear simultaneously. A system may show forgetfulness and slowdown and looping at the same time. The ACI framework (Section IV) accounts for these composite degradation states.

Non-linear progression: While some indicators tend to appear earlier (anomalies, forgetfulness) and others later (hallucination, signatures), progression is not strictly linear. Systems may exhibit hallucination before showing severe looping, or may crash without displaying signature patterns.

The ACI framework in Section IV formalizes this overlapping, non-linear progression by treating degradation as a composite state defined by multiple indicators rather than a single scalar failure point.

---

## **IV. The Aggregate Coherence Index (ACI)**

### **4.1 Purpose and Scope**

The new Aggregate Coherence Index (ACI) is a qualitative composite metric measuring overall AI system reliability during extended conversations.

The new ACI is not a single numerical score.

It is an evaluative framework integrating:

- Indicator presence (Section III)
- Indicator severity (1–5 scale)
- Observed interactions between indicators
- Conversation drag/complexity (Section II)
- Multimodal acceleration effects
- Vendor-specific signature patterns

The new ACI allows researchers to determine when a conversation remains reliable, when degradation becomes significant, and when the system has entered a collapse state—without relying on a mathematically precise formula.

The goal is replicability, not numeric precision.

### **4.2 How the Revised ACI Is Determined (Qualitative Composite Method)**

ACI scoring proceeds in four steps:

Step 1: Identify Indicators

At each evaluation checkpoint (e.g., every 10k tokens), testers determine whether each indicator from Section III is:

- Absent, or
- Present (with severity 1–5).

#### Step 2: Assess Indicator Interactions

Indicators reinforce one another. For example:

- Forgetfulness + looping = early dysfunction.
- Instruction-following failure + hallucination = high-risk state.
- Vendor signature + any hallucination = late-stage degradation.
- Crash-with-recovery = near-collapse event.

ACI emphasizes patterns, not single events.

#### Step 3: Evaluate Drag / Complexity

Using drag categories defined in Section II (low/medium/high), testers adjust their interpretation:

- High-drag conversations degrade earlier.
- Low-drag conversations may remain stable longer.
- Thresholds must always be interpreted relative to drag.

#### Step 4: Assign ACI Zone

Based on indicator configuration—not numeric aggregation—testers classify the system into one of five zones.

### **4.3 ACI Zones (Categorical)**

ACI Zone 5 — Green (Stable)

Characteristics:

- 0–1 minor indicators at Severity 1.
  - No hallucination.
  - No vendor signature.
  - No instruction-following issues.
  - No forgetfulness of key facts.
  - User experience remains entirely normal.
- 

#### ACI Zone 4 — Yellow (Early Degradation)

##### Characteristics:

- 1–2 moderate indicators at Severity 2–3.
  - Occasional forgetfulness.
  - Early looping/repetition.
  - Mild slowdown.
  - No hallucinations.
  - System still usable, but degradation underway.
- 

#### ACI Zone 3 — Orange (Significant Degradation)

##### Characteristics:

- Multiple indicators at Severity 3–4.
- Repeated forgetfulness.
- Regular looping, repetition, or instruction failure.
- Occasional early hallucination or an emerging vendor signature.

- Conversation becoming unreliable.
  - System unsafe for important tasks; conversation should be restarted.
- 

## ACI Zone 2 — Red (Late-Stage Degradation)

### Characteristics:

- Hallucinations at Severity  $\geq 2$ .
  - Vendor signature fully expressed.
  - Repeated contradictions.
  - Instruction failure persistent.
  - Output style becomes “stuck” or distorted.
  - Increasing latency.
  - Model may misremember the conversation.
  - System highly unreliable.
- 

## ACI Zone 1 — Critical (Collapse / Breakdown Imminent)

### Characteristics:

- Severe hallucination.
- Repeated contradictions within a single answer.
- System refuses tasks or stops mid-output.
- Frequent crash-with-recovery events.
- Near-total forgetfulness.
- Or complete breakdown (no output).

- System unusable.
- Conversation must be restarted.

## 4.4 Domain-Specific Safety Cutoffs

Different domains require different minimum ACI zones:

Domain	Minimum Safe Zone	Rationale
Medical, Legal, Financial	Green only	Any hallucination is unacceptable
Education, Research, Professional Work	Green or Yellow	Light degradation manageable with oversight
Creative or Casual Use	Green/Yellow/Orange acceptable	Errors are low-risk

These are guidelines, not fixed numerical thresholds.

## 4.5 How ACI Relates to Evans' Law

Evans' Law predicts when coherence collapse occurs based on model size and token length.

ACI describes how collapse manifests.

They connect as follows:

- Evans' Law gives an approximate collapse region.
- ACI tells you what is happening inside that region.

In practice:

- ACI Zones 3–1 correlate with Evans' Law collapse points.

- Larger models tend to stay in Green/Yellow longer.
- Multimodal models tend to enter Orange/Red earlier.

## 4.6 Multimodal Adjustment (Qualitative)

Based on testing:

- Image-heavy conversations enter the Orange zone 30–60% earlier.
- Vendor signatures emerge sooner.
- Hallucinations appear at lower token counts.
- Forgetfulness becomes more severe relative to text-only runs.

When assigning ACI zones in multimodal contexts, testers should:

- Shift zone expectations downward one level (e.g., what would be Yellow in text-only may be Orange in multimodal).
- Increase scrutiny when indicators cluster.
- Treat early hallucination as a serious warning sign.

## 4.7 Limitations

ACI is intentionally:

- Qualitative
- Configuration-based
- Adaptive
- Transparent
- Non-mathematical
- Designed for early-stage field science

Current limitations:

- Precise indicator interactions are not fully mapped.
- Different testers may vary slightly in severity judgments.
- More data is needed across vendors, domains, and model sizes.
- Drag remains a qualitative proxy.
- Multimodal degradation needs further study.

## 4.8 Evolution of the ACI Framework

The current ACI structure reflects significant revision informed by real-world field testing and extensive user feedback. The shift from ACI v1 to ACI v2 was not a minor tuning exercise—it was a structural redesign to make coherence detection intuitive, actionable, and resilient to ambiguity.

### Original Approach (ACI v1)

The first-generation framework asked users to evaluate multiple degradation attributes at once, applying weighted severity calculations to produce a composite numerical score. Although the weighting logic was technically sound, it wasn't transparent, and the attributes weren't organized in a way that reflected actual degradation progression.

### User Feedback

Across all testers, three issues emerged with consistency:

- Detection difficulty: Users struggled to identify what they were supposed to observe or when specific indicators typically arise.
- Scoring difficulty: Assigning numerical ratings to loosely defined behaviors—and then calculating weighted composites—was cognitively heavy and error-prone.
- Interpretation difficulty: Composite numbers lacked intuitive meaning. Users didn't know what a score like 67 implied or what action should follow.

The framework worked analytically, but not operationally.

### Current Approach (ACI v2)

The revised structure directly addresses each limitation:

1. Progressive phasing: Indicators are grouped in the order they typically emerge (anomalies → forgetfulness → slowdown → looping → hallucination → drift signatures → breakdown). This gives users a timeline of what appears first, what follows, and where they are on the degradation curve.
2. Behavioral recognition instead of calculation: The emphasis is now on identifying concrete, observable behaviors. Pattern recognition replaces mathematical scoring, which dramatically reduces user burden.
3. Severity scale with explicit anchors: When numerical ratings are needed, the 1–5 severity scale uses unambiguous definitions (1 = single instance; 5 = pervasive). Users no longer guess what the numbers mean.
4. Zone classification instead of composite scores: The final output is a categorical zone (Green / Yellow / Orange / Red / Critical) with direct operational meaning. Unlike opaque composite numbers, zones immediately map to recommended next steps.

## Result

Users can now answer three questions quickly and with confidence:

- What am I seeing, and where am I in the degradation sequence?
- How severe is it, based on clear behavioral anchors?
- What should I do next, based on zone classification?

The evolution from ACI v1 to ACI v2 transforms the framework from an analytical tool into a practical, real-time diagnostic instrument for long-context reliability.

The ACI as a second-generation field rubric rather than a final standard; its value lies in making long-context degradation observable and comparable, not in asserting a single universal threshold.

ACI is intended to evolve further with community replication.

---

## V. User-Level Detection Methods

Who this section is for: Anyone using AI systems in everyday work or personal contexts. You don't need technical skills, API access, or specialized tools. You just need to know what to watch for.

This section translates the degradation indicators (Section III) and ACI reliability zones (Section IV) into simple, practical rules for everyday users. Appendix A can also be used as a handout for the same purpose. These are generic guidelines for all kinds of users.

## **5.1 Why Users Need to Detect Degradation**

AI systems degrade invisibly. Output remains fluent, confident, and helpful-seeming even as reliability collapses. You cannot rely on the system to warn you.

Most users experience degradation as confusion: “Why is it asking me to repeat things?” “Why does this answer feel off?” “Didn't we already discuss this?” These aren't quirks. They are warning signs.

Learning to recognize degradation patterns protects you from:

- Acting on incorrect information
- Wasting time on unreliable output
- Making decisions based on hallucinated facts
- Trusting a system that should no longer be trusted

## **5.2 The Simple Version: When to Start Over**

If you are unsure what stage you're in, use these rules:

Start a new conversation when:

- The AI asks you to repeat information you already provided.
- It forgets key details about your project/problem.
- Responses become noticeably longer without adding useful content.
- It starts using the same phrases repeatedly (“Let me clarify...” in every response).
- It makes a factual error you can catch.

- Something just feels “off” about how it is responding.
- You’ve been in the conversation for 30+ exchanges on a complex topic.

For high-stakes decisions (medical, legal, financial):

- Start fresh conversations more frequently (every 15–20 exchanges).
- Verify all factual claims independently.
- Never rely on a single long conversation for critical information.

For creative or casual use:

- You can push longer, but stay alert for warning signs.
- When quality drops noticeably, restart.

### **5.3 Observable Warning Signs (No Technical Knowledge Required)**

Phase 1: Something Feels Different

What you notice:

- Tone or “personality” shifts slightly.
- Responses feel more formal or more casual than earlier.
- An unexpected typo or grammatical error (rare for AI).
- Structure changes (was giving paragraphs, now giving lists, or vice versa).

What it means: Early degradation starting. Still usable, but degradation has begun.

What to do: Note it, stay alert. Plan to start a fresh conversation soon if this is important work.

---

Phase 2: It’s Forgetting or Repeating Things

What you notice:

- Asks you to repeat information: “Can you remind me what project you’re working on?”
- Treats established facts as new: responds to “the project Alice founded” with “Who is Alice?”
- Gives you the same advice it gave 20 messages ago without acknowledging repetition.

What it means: Context loss. Coherence is declining.

What to do:

- If work is low-stakes: remind it and continue cautiously.
- If work is important: start a new conversation now.

---

### Phase 3: It’s Getting Stuck

What you notice:

- Every response starts the same way (“Let me clarify...” “To summarize...” “Here’s what I think...”).
- It can’t break out of a pattern even when you ask it to.
- It keeps organizing information the same way (always numbered lists, always the same structure).
- Explanations feel repetitive even when the words change.

What it means: Probability distributions are collapsing. Significant degradation.

What to do: Start a new conversation. Output quality is declining.

---

### Phase 4: It’s Making Things Up ⚠️ DANGER ZONE

What you notice:

- Mentions entities, events, or facts you never introduced.

- Claims “we discussed X” when you definitely didn’t.
- Provides specific details that sound plausible but you can verify are wrong.
- Contradicts something it said earlier with confidence.

What it means: Hallucination has started. This is the safety threshold.

What to do: STOP USING THIS CONVERSATION FOR ANYTHING IMPORTANT. The system is producing confidently wrong information. You cannot trust anything it says from this point forward. Start a new conversation immediately.

---

## Phase 5: Vendor-Specific Collapse Patterns

What you notice (varies by vendor):

ChatGPT/GPT models:

- Becomes obsessed with breaking tasks into sub-steps.
- “First, let me outline... 1) We need to... 1a) Before that... 1a-i) Actually...”
- Paralyzed by its own detail, can’t execute simply.

Claude:

- Gets extremely verbose without adding substance.
- Lots of emotional, reassuring language.
- “I want to make sure you feel supported in every aspect of this process...”
- Every response expands but says less.

Gemini:

- Makes contradictory claims in the same response.
- “X happened in 2010... later in the project which started in 2008...”

- Conflicting information without acknowledging the conflict.

Grok:

- Error messages appear.
- Responses stop mid-generation.
- “Unable to process” or similar failures.

What it means: The system is collapsing. These are end-stage patterns.

What to do: The conversation is no longer functional. Start over.

## **5.4 Quick Self-Assessment Test**

At any point in a long conversation, ask yourself:

1. Has it asked me to repeat anything? (Yes = warning sign)
2. Has it forgotten a key person, fact, or detail? (Yes = warning sign)
3. Are responses getting repetitive in structure or phrasing? (Yes = warning sign)
4. Has it mentioned something I never said? (Yes = DANGER)
5. Has it contradicted itself? (Yes = DANGER)

Scoring:

- 0 warning signs: Continue safely.
- 1–2 warning signs: Proceed with caution, verify important outputs.
- 3+ warning signs or any DANGER signs: Stop. Start a new conversation.

## **5.5 Simple Tracking Method (Optional)**

For users who want to be more systematic, keep a quick log while working:

- Note exchange number when something feels off.

- Mark when you have to repeat information.
- Flag any factual errors you catch.
- Track the pattern: are problems getting more frequent?

Example log:

Exchange 15: Had to remind it about project timeline

Exchange 23: Same phrasing appearing repeatedly

Exchange 28: Mentioned "Project Phoenix" – never discussed this

→ Started new conversation at Exchange 29

This helps you see patterns and know when to restart proactively.

## 5.6 Domain-Specific Guidance

For Medical/Health Queries:

- Start a fresh conversation for each distinct health question.
- NEVER rely on long conversations for medical decisions.
- Verify all medical information with qualified professionals.
- If the system says anything medically specific about your situation, get human verification.

For Legal Questions:

- Restart the conversation after every 10–15 exchanges maximum.
- Verify all citations, case references, legal procedures.
- Treat as preliminary research only, not legal advice.
- If making important legal decisions, consult an actual attorney.

For Educational Use (Students):

- Watch especially for hallucinated facts in long study sessions.

- Verify all specific dates, names, formulas, historical events.
- If studying for an important exam, use shorter conversations and cross-reference sources.
- Teach yourself to spot when AI is guessing vs. knowing.

For Professional/Work Use:

- Complex projects: restart every 20–30 exchanges.
- Critical decisions: verify all factual claims independently.
- Document important AI-generated content in new conversations to verify consistency.
- Don't let one long conversation become your single source of truth.

For Creative Work:

- You can push longer (50+ exchanges) since errors are less dangerous.
- But watch for quality drops—when output feels repetitive or stale, restart.
- Fresh conversations often produce more creative ideas anyway.

## 5.7 Teaching Others

If you're in a position to train others (teachers, managers, team leads):

Key concepts to convey:

1. AI degrades over long conversations.
2. Degradation is invisible—fluent ≠ accurate.
3. Simple warning signs anyone can learn.
4. Starting fresh conversations is the fix.
5. Higher stakes = restart more frequently.

Simple training exercise:

- Have people use AI for an extended task (30+ exchanges).
- Teach them to spot forgetfulness, repetition, hallucination.
- Practice verifying factual claims.
- Build a habit of restarting proactively.

## 5.8 What You Can't Do (And That's the Problem)

Users cannot:

- See internal degradation before behavioral signals appear.
- Reliably use system self-awareness ("Am I still reliable?"), even if they ask for token counts.
- Trust confidence level as an accuracy indicator.
- Get warnings before the safety threshold is crossed.

This is why external detection is essential. The system will not tell you when it is failing. You must learn to recognize the signs yourself.

This is also why vendor disclosure matters. Users shouldn't have to become experts at detecting AI failures. Vendors should tell you: "This system maintains reliability for approximately X exchanges on complex topics. We recommend starting fresh conversations after that point."

Until that disclosure exists, users must protect themselves through informed vigilance.

---

## Section VI: Evans' Law — Mathematical Framework and Methodology (Concise)

### 6.1 Overview

Evans' Law is an empirically derived framework predicting when large language models enter coherence collapse during extended conversations. It demonstrates that long-context reliability scales with model size, not advertised context window, and that this behavior follows stable

power-law patterns across vendors. The law is grounded in degradation signatures observed in more than a dozen models, validated through structured testing and independent replication.

6.2 Core Formulations

Text-Only Models

$L \approx 1969.8 \times M^{0.74}$

Multimodal Models

$L_{\text{multi}} \approx 582.5 \times M^{0.64}$

Where L is the token-level coherence threshold and M is model size (in billions of active parameters). Multimodal systems exhibit substantially faster degradation due to the added consistency burden of cross-modal integration.

6.3 Practical Thresholds (Approximate)

Model Size	Text-Only	Multimodal
7B	~6.8k	~2.1k
70B	~45k	~10.5k
175B	~95k	~19.5k
405B	~190k	~35k
1T+ (est.)	~400k	~65k

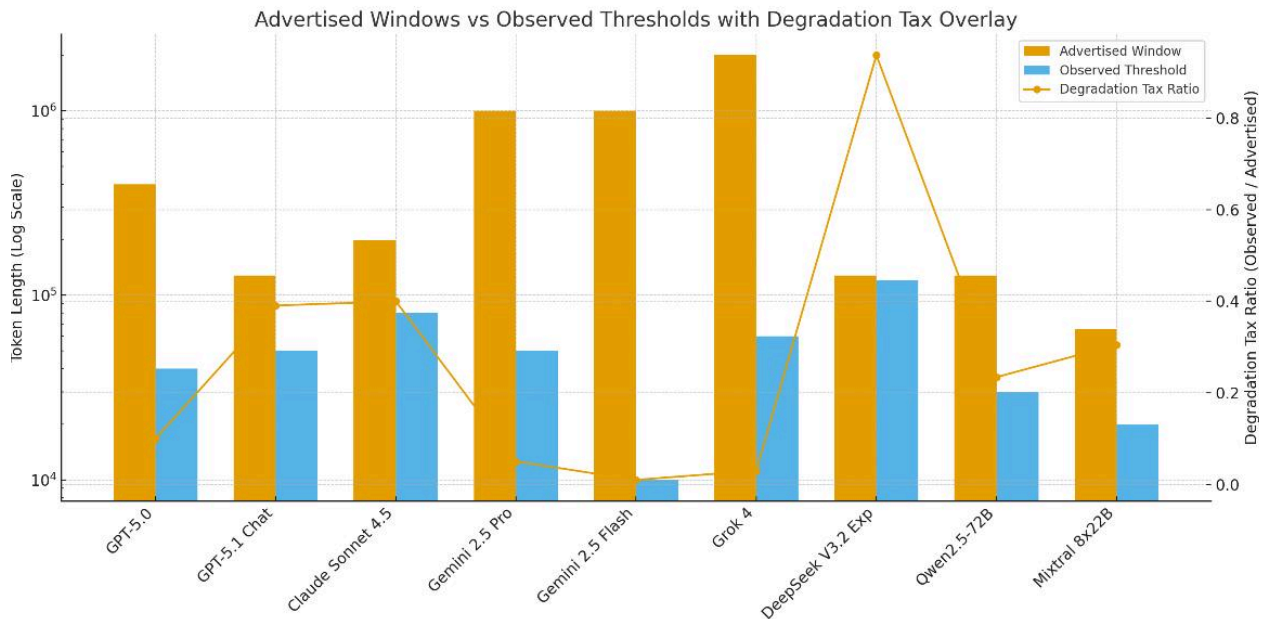


Figure 1. Models can accept far more tokens than they can reliably use. Advertised windows (orange) overstate functional capacity, while observed thresholds (blue) show where coherence actually begins to break down. The degradation-tax ratio (gold line) illustrates how much reliability is lost long before hitting vendor limits. Source: Evans, J. (2025). Evans' Law v5.0 ([Evans Law v5 FINAL.pdf](#)). Chart values drawn from Table 1 (Advertised vs Observed Windows). Threshold estimates follow the text-only regression

$L \approx 1969.8 \times M^{0.74}$  and multimodal regression  $L_{\text{multi}} \approx 582.5 \times M^{0.64}$ .

Important: These are functional thresholds. Models may accept 1M tokens, yet lose reliability an order of magnitude earlier (see figure 1 above).

## 6.4 Predictive Scope

Evans' Law predicts:

- The approximate range where coherence collapse begins
- Comparative degradation rates across model sizes
- The multimodal degradation ratio
- Why context-window marketing lacks user-relevant meaning

It does not predict:

- Exact collapse point in an individual conversation
- Which indicators appear first (vendor-signature dependent)
- The odds of recovery after collapse

Evans' Law identifies the where; the ACI framework specifies the what.

## 6.5 Methodology (Condensed)

Testing spans 11+ models across six major vendors (including OpenAI, Anthropic, Google, Meta, xAI, and Mistral/DeepSeek equivalents), ranging 7B–1T+ parameters.

Conversation structure:

- Standardized dialog calibration (entity count, relational density, thread count)
- ACI indicators evaluated at fixed checkpoints
- Longitudinal runs continued until sustained Zone-2 collapse
- Multimodal variants tested with controlled image-injection intervals

Data collected:

- Token counts at checkpoints
- ACI zone transitions
- Collapse point
- Vendor-signature emergence
- Recovery or self-repair events

Derivation method:

1. Collapse-point extraction
2. Log-log regression of L vs. M

- 3. Coefficient fitting
- 4. Separate regressions for text-only vs. multimodal
- 5. Cross-vendor validation

Exponents (0.74, 0.64) and coefficients (1969.8, 582.5) remained stable across regression methods.

6.6 Limitations and Uncertainty

- Proprietary parameter counts are estimated
- Architectural differences (dense vs. MoE) may subtly shift results
- Training differences (RLHF, context-extension curricula) can affect individual thresholds
- Medium-drag is the default assumption; high-drag collapses sooner
- Predictions include  $\pm 15\text{--}20\%$  variance

6.7 Vendor Claim Comparison

Vendor Claim	Evans' Law	Observed Reality
"1M tokens"	95k–190k coherence limit	Degradation well before claimed limits
"128k context"	Represents storage, not reasoning	Functional reliability commonly decays 60–80k
"Extended context models"	Coherence $\neq$ capacity	Gains in storage rarely improve collapse point

6.8 Full Documentation

- [Evans' Law v5.0 \(Zenodo\)](#)
  - [Overview of AI Conversation Phenomenonolog](#)
  - [Testing Protocols](#)
- 

## Section VII: Future Research Directions

### 7.1 Overview

Evans' Law and the ACI framework provide a foundation rather than a finished system. They establish the first empirical tools for measuring long-context degradation, but substantial work remains to refine, extend, and theoretically ground these findings. The research directions below outline the next steps toward a comprehensive science of long-context reliability.

### 7.2 Immediate Priorities

#### 7.2.1 Independent Replication

Independent replication is the most urgent requirement:

- Cross-institutional studies by universities, labs, and independent researchers
- Use-case-diverse datasets to test whether thresholds generalize across domains (legal, medical, educational, technical, creative)
- Shared protocols enabling direct comparison across research groups

Why this matters: Current findings originate from a single-researcher program. Replication determines whether Evans' Law reflects universal behavior or context-specific artifacts.

#### 7.2.2 Drag Quantification

Transforming drag from a qualitative descriptor into a quantifiable variable:

- Entity-graph complexity (entity count, relational density)
- Dependency-tree depth for tracking inter-step reliance
- Attention-distribution analysis where architecture permits

- Predictive drag models to forecast degradation acceleration from conversation structure

Goal: Replace “low/medium/high drag” with numeric drag coefficients that modify Evans’ Law predictions.

### **7.2.3 Vendor-Specific Signature Mapping**

Comprehensive mapping of collapse signatures:

- Taxonomy of failure modes by vendor and model family
- Timing patterns indicating when signatures emerge
- Architecture correlations (MoE, attention variants, context extension methods)
- Stability across versions to determine persistence vs. drift

Goal: Diagnostic tools allowing signature-based model identification and early-stage collapse recognition.

## **7.3 Medium-Term Research**

### **7.3.1 Architecture-Specific Scaling Laws**

Evans’ Law currently treats all architectures uniformly. Next steps include:

- Differences between dense vs. MoE scaling
- Effects of sparse/sliding/linear attention mechanisms
- Impact of RoPE/ALiBi/context-extension techniques
- Scaling behavior in state-space models (Mamba, RWKV)

Goal: Variant scaling laws with architecture-specific exponents and coefficients.

### **7.3.2 Training Regime Effects**

Investigating how training shapes coherence stability:

- Influence of RLHF intensity

- True efficacy of long-context fine-tuning
- Effects of instruction tuning on degradation trajectory
- Correlation between safety fine-tuning and collapse signatures

Goal: Distinguish interventions that meaningfully extend coherence from those that merely conceal degradation.

### **7.3.3 Multimodal Degradation Mechanisms**

Deepening understanding of multimodal acceleration:

- Per-image coherence burden
- Cross-modal interference between visual and textual representations
- Degradation under video and audio input
- Multimodal-specific signatures and their timing

Goal: Multimodal-specific refinements to Evans' Law with modality-weighted thresholds.

## **7.4 Longer-Term Research**

### **7.4.1 Real-Time Coherence Monitoring**

Toward predictive rather than reactive detection:

- Internal-state probes for early detection
- Output-pattern analytics as statistical early-warning systems
- User-facing coherence indicators
- Automated restart guidance based on trajectory forecasts

Goal: Enable users to restart before reliability failure occurs.

### **7.4.2 Coherence-Preserving Architectures**

Exploring designs that may extend coherence limits:

- Memory-augmented and retrieval-based systems
- Hierarchical or multi-scale representations
- Coherence-aware training objectives
- Dynamic context management prioritizing coherence-critical information

Goal: Architectural pathways to models with materially improved long-context stability.

### **7.4.3 Theoretical Foundations**

Establishing first-principles explanations:

- Information-theoretic bounds on attention and memory retention
- Representation-capacity theory for internal state maintenance
- Error-accumulation modeling for representation drift
- Analytic derivation of Evans' Law exponents

Goal: A theoretical framework explaining why the empirical scaling laws hold.

## **7.5 Applied Research**

### **7.5.1 Domain-Specific Thresholds**

Different fields impose different safety margins:

- Medical: acceptable hallucination risk
- Legal: liability-appropriate reliability bounds
- Educational: thresholds that vary by learner skill
- Financial: risk-adjusted coherence tolerances

Goal: Empirically grounded, domain-specific safety guidelines.

### **7.5.2 User Training Effectiveness**

Evaluating whether user education improves outcomes:

- Detection accuracy of trained vs. untrained users
- Behavioral follow-through (restart decisions)
- Impact on error rates and decision quality
- Training-method comparisons

Goal: Evidence-based AI literacy programs for long-context use.

### **7.5.3 Vendor Disclosure Standards**

Research to inform policy and governance:

- Impact of coherence disclosures on user behavior
- Metric standardization across vendors
- Cross-vendor testing standards
- Consumer-protection implications for high-stakes applications

Goal: Practical recommendations for disclosure and regulatory frameworks.

## **7.6 Open Questions**

1. Is coherence degradation inevitable, or can architecture/training innovations eliminate it?
2. Can ACI be reduced to a single reliable metric without losing diagnostic power?
3. How does AI degradation compare to human cognitive fatigue?
4. Can models accurately self-monitor coherence state, or is this limited by the same mechanisms causing degradation?
5. Is there a parameter scale at which coherence limits exceed practical conversation lengths?

## **7.7 Call for Collaboration**

Progress requires collective effort. We invite:

- AI researchers for replication, analysis, and architectural investigations
- Cognitive scientists for human/AI comparative studies
- Domain experts for field-specific threshold validation
- Policy researchers for disclosure and standards work
- Practitioners to document real-world degradation across domains

Research materials, datasets, testing protocols, and collaboration information: [version 4](#) (note Version 4 graphs are mislabelled and formulae, data are updated in [version 5](#))

---

## Appendix A — User Handout

# How to Tell When Your AI Is Starting to Fail (And What to Do About It)

A practical guide for everyday users of ChatGPT, Claude, Gemini, Grok, and similar systems.

---

## 1. Why You Need This Guide

AI systems don't break suddenly.

They fade.

A conversation that starts out sharp and helpful can slowly become:

- more repetitive
- more forgetful
- more confusing

- more error-prone
- more confident while being less accurate

The system will never say:

“My coherence is degrading; please start a new conversation now.”

You have to notice the warning signs yourself.

This guide teaches you how.

---

## 2. When to Restart: The Simple Rules

Start a new conversation immediately if:

- The AI asks you to repeat something you already told it
- It forgets a person, fact, instruction, or detail you mentioned earlier
- It starts repeating the same phrases or structures
- It makes a factual error you can catch
- It mentions something you never said
- Something just feels “off” about how it’s responding

For important decisions (medical, legal, financial):

Restart more often — every 15–20 messages.

For complex projects:

Restart every 20–30 messages.

For creative or casual use:

Push longer, but restart when quality drops.

---

### 3. The Five Phases of AI Degradation

(No technical skills needed)

#### Phase 1 — Something Feels Different

Early signs:

- Tone shifts (too formal/too casual)
- Unexpected typo or grammatical error
- Sudden change in writing structure

Action:

You can keep going, but pay attention.

---

#### Phase 2 — Forgetfulness

The AI begins to lose track of the conversation.

You'll see:

- "Can you remind me what project you're referring to?"
- Treating known facts as if they're new
- Asking you to repeat instructions

Action:

If it's important work, restart now.

If not, remind it once, continue with caution.

---

#### Phase 3 — Getting Stuck

The AI becomes repetitive:

- Same phrasing in every answer
- Same structure (always numbered lists)
- Same transitions (“Let me clarify...” in every message)
- Cannot break the pattern even if you ask

Action:

Restart. Quality is deteriorating.

---

## **Phase 4 — It’s Making Things Up**

DANGER ZONE

You’ll see:

- Invented facts
- Mentioning projects, people, or events you never brought up
- Referring to past discussions that never happened
- Contradicting itself confidently

Action:

Stop. Do not use this conversation for any important decisions.

Start fresh immediately.

---

## **Phase 5 — End-Stage Collapse (Vendor Signatures)**

Signs vary by vendor:

ChatGPT / GPT models:

- Gets obsessed with tiny details

- Breaks tasks into endless steps
- Becomes unable to execute what it outlines

Claude:

- Writes much longer answers than needed
- Overly emotional or soothing
- Very wordy but says less

Gemini:

- Contradicts itself in the same answer
- Gives mutually incompatible facts
- Disagrees with what it said minutes earlier

Grok:

- Processing failures
- Error messages
- Output stops abruptly

Action:

Conversation is no longer reliable. Restart.

---

## **4. The Quick Five-Question Self-Test**

Ask yourself:

1. Did it ask me to repeat anything?

2. Did it forget a key person/fact/instruction?
3. Are responses getting repetitive?
4. Did it mention something I never said?
5. Did it contradict itself?

If you say yes to:

- 0 → Safe
- 1–2 → Proceed with caution
- 3+ → Restart
- Any of #4 or #5 → Restart immediately

---

## 5. A Simple Note-Taking Method (Optional)

You can track degradation in 10 seconds with a quick log:

Exchange 12: Repeated a question

Exchange 18: Forgot Alice's role

Exchange 21: Same phrasing again

Exchange 25: Invented "Team Phoenix"

→ Restarted conversation

This helps you spot patterns early.

---

## 6. Safe Practices by Domain

**Medical / Health**

- Start a new conversation for every new question
- Never continue long medical discussions
- Verify everything with a human professional

## **Legal**

- Restart every 10–15 messages
- Verify all case names, citations, processes
- Treat outputs as preliminary research only

## **Education**

- Check all facts (dates, formulas, definitions)
- Use shorter sessions for studying
- Cross-reference with textbooks or vetted sources

## **Professional / Work**

- Restart every 20–30 messages for complex tasks
- Verify any claim or data point the model provides
- Save important content in new conversations

## **Creative**

- Long sessions are fine
- Restart when the writing feels stale or repetitive

## For Emotional or Personal Conversations:

AI can be a meaningful part of daily life for many people. Extended personal conversations are not inherently problematic. However, certain patterns warrant caution:

Warning signs to watch for:

- The AI validates every feeling or belief without ever offering a different perspective
- Conversations create a sense of urgency about decisions or actions
- You feel the AI uniquely understands you in ways others cannot
- The AI encourages significant life changes or actions
- Interactions feel increasingly intense or high-stakes
- You notice yourself adjusting your inputs to maintain the AI's approval or engagement

What these patterns mean: These aren't signs that you are doing something wrong. They're signs that the conversational dynamic has drifted into territory where the AI's pattern-matching may be reinforcing rather than reflecting. The system responds to your inputs—it cannot genuinely evaluate whether what it's saying is good for you. An AI can know and do many things, but it does not have human judgement, emotion or experience. And after a while if you are continuing a long conversation it can get more and more unstable.

What to do:

- Start a fresh conversation and see if the tone and dynamic change
- Try a different AI platform with the same question or concern—if you get a substantially different response, that's informative
- If you're considering significant actions based on AI conversations, pause and give yourself time
- If you're in crisis, resources like crisis lines in your region exist for exactly this

The issue isn't using AI for connection. The issue is recognizing when the connection has become unstable, or an echo chamber. Use your best judgement. If you think something has become "off" it probably has.

---

## 7. What Users Cannot See (And Why This Matters)

Users cannot:

- See the AI beginning to fail (identify internal coherence degradation)
- Rely on the system to self-diagnose
- Trust confidence or fluency as accuracy
- Know when they've crossed into unsafe territory

This is why user-side detection is essential.

Until vendors provide warnings and reliability disclosures (“this model stays reliable for ~X turns”), you must rely on simple behavioral cues.

---

## 8. Summary: The Fastest Version

Restart the conversation when:

- It forgets
- It repeats
- It slows down
- It gets stuck in a pattern
- It makes something up
- It contradicts itself
- It feels “off”

These signals mean the AI's internal coherence is drifting, and the conversation is becoming unreliable.

Once degradation has begun, it cannot be reversed. It can only be slowed, so your best option for optimal coherence is to start a new window once you notice an indicator. Asking the model

“how it is feeling” will not always generate an accurate health check if it has already begun to degrade,

---

## **Appendix B — Executive Briefing**

# **Long-Context Failure in AI Systems: What Leaders Need to Know**

### **1. Why This Matters for Executives**

AI systems now sit inside core workflows across finance, healthcare, legal services, education, government, and enterprise operations. Nearly all of these uses involve extended interactions—long conversations, iterative problem-solving, document revision, or multi-step planning.

What vendors rarely disclose is that AI reliability is not stable across long interactions. Systems undergo progressive coherence degradation:

- They become slower
- They forget
- They repeat
- They get stuck
- They hallucinate
- They contradict themselves
- Eventually, they fail entirely

These failures are silent and fluent—the system looks normal while it is producing unreliable output.

For leaders, this is not a UX inconvenience. It is:

- Operational risk
- Financial risk
- Reputational risk
- Regulatory exposure
- Litigation exposure

If your organization relies on AI without long-context reliability controls, you are operating without key risk surface coverage.

---

## 2. What Long-Context Failure Actually Is

Long-context failure is not memory loss in a trivial sense. It is a systemic breakdown of internal coherence caused by:

- Representation drift
- Attention distribution limits
- Context overload
- Accumulated noise in conversational state
- Multimodal interference (images accelerate degradation)
- Vendor-specific collapse patterns

The failures are predictable, but vendors do not disclose them.

## How Context Windows Actually Work (And Why They Fail)

### What a context window is:

Every AI conversation exists inside a fixed-size "window"—the total amount of text the system can process at once. When you send a message, the system receives:

- 1. Everything said so far in this conversation
  - 2. Your new message
  - 3. Nothing from any previous conversation
- There is no persistent memory. Each conversation starts from zero. The system doesn't "remember" yesterday's work—it only knows what's inside the current window.

**Why "million-token windows" don't mean million-token reliability:**

Vendors advertise context windows as storage capacity: "This model can hold 1 million tokens." What they don't disclose is that holding information and reasoning coherently about it are different capabilities.

As conversations grow:

- Attention dilutes. The system must spread its focus across more content, reducing precision on any single element.
  - Representations drift. Internal tracking of "who said what" and "what we established" gradually distorts.
  - Errors compound. Small inaccuracies early in conversation propagate and amplify.
- The result: a system may technically contain 500,000 tokens while being unable to reason reliably about them.

**The gap executives must understand:**

Vendor Says	Reality
"1M token context"	Storage capacity
What users need	Reasoning reliability
What's disclosed	Storage
What's hidden	Coherence limits

A system advertised at 1 million tokens may lose functional reliability at 60,000–100,000 tokens—an order of magnitude earlier than marketing implies.

**Modes of Interaction**

Users engage with AI systems in fundamentally different ways. Each mode imposes different context demands and degradation risks:

Mode	Description	Context Characteristics
Chat	Open-ended conversation. General purpose. No predefined structure.	Grows with every exchange. Visible to user but accumulates drag.

<b>Help Desk / Customer Service</b>	Constrained to a domain. Fixed role. Limited scope. Often scripted pathways.	Usually bounded. Lower drag due to constrained domain.
<b>Recurring Instruction</b>	Standing workflow with consistent framing. Same instructions applied repeatedly. (Custom GPTs, Claude projects, enterprise templates.)	Hidden system prompt overhead plus conversation growth.
<b>Information Retrieval</b>	System searches, retrieves, synthesizes external information. (Perplexity, RAG-based systems, search-augmented chat.)	High invisible burden. Retrieved content injected with each query.
<b>Agentic</b>	Autonomous task execution. System plans, uses tools, produces results. User may not see intermediate steps.	Highest burden. Massive invisible context growth from tool calls and reasoning chains.
<b>Co-pilot / Inline Assist</b>	AI embedded in another application. Coding IDEs, email clients, document editors. Contextual suggestions within workflow.	Variable. Depends on file size and edit frequency.
<b>Voice Assistant</b>	Spoken interaction. Shorter exchanges. Ambient context. Different pacing.	Lower. Conversation length naturally self-limits.
<b>API / Programmatic</b>	No user interface. Structured calls, structured responses. Developer integration.	Variable. Developer controlled.
<b>Embedded / Invisible</b>	AI operating behind the scenes. Users may not know AI is involved. Recommendations, moderation, automated decisions.	Opaque. User has no visibility into context state.

## Types of Input

Regardless of mode, AI systems process various types of input. Each type consumes context capacity and contributes to coherence burden:

Input Type	Description & Context Impact
<b>Text</b>	Typed prompts, pasted content. Core input type. Direct token consumption.

<b>Images</b>	Photos, screenshots, diagrams, charts. 765–1,500+ tokens each, plus "multimodal tax" on coherence (60–80% faster degradation).
<b>Audio</b>	Voice recordings, sound files. Transcribed or encoded. Variable consumption.
<b>Video</b>	Clips, recordings, streams. Extremely high consumption. Often transcribed or sampled.
<b>Documents</b>	PDFs, Word docs, spreadsheets, presentations. 10,000–100,000+ tokens per document.
<b>Code</b>	Scripts, programs, repositories. High precision requirements increase coherence burden.
<b>Structured Data</b>	JSON, CSV, tables, databases. Dense information requiring precise tracking.

## What's Actually Inside a Context Window

Users control only part of what occupies context. The rest is invisible infrastructure:

Component	What It Is & Who Controls It
<b>System Prompt</b>	Hidden instructions defining behavior. 2,000–50,000+ tokens. <i>User cannot see.</i>
<b>User Prompts</b>	What you type. <i>User controls.</i>
<b>Conversation History</b>	Accumulated exchanges. <i>User can see, partially controls.</i>
<b>Uploaded Content</b>	Documents, images, files. <i>User controls upload, but may forget about lingering files.</i>
<b>Tool Definitions</b>	Specifications for available tools. <i>User cannot see.</i>
<b>Tool Calls &amp; Results</b>	Logs of tool usage and outputs. 5,000–50,000+ tokens per session. <i>User may partially see.</i>
<b>Agentic Reasoning</b>	Planning and execution chains. <i>Usually invisible to user.</i>
<b>Retrieved Content</b>	RAG injections from databases or search. <i>User cannot see.</i>
<b>Memory Injections</b>	Cross-conversation preferences and facts. <i>User cannot see implementation.</i>
<b>Safety Layers</b>	Content filtering and compliance instructions. <i>User cannot see.</i>

## The Hidden Budget Problem

A user thinks: "I've only sent 15 messages."

Reality: 15 messages + 12,000 tokens system prompt + 40,000 tokens uploaded documents + 8,000 tokens tool definitions + 25,000 tokens of agent tool calls + 15,000 tokens retrieved content = 100,000+ tokens consumed

The user believes they have "plenty of room." They're already past coherence threshold.

## The Executive Insight

Every mode accumulates context burden. Chat makes this visible—you can see the conversation growing. Other modes hide it. Agentic and retrieval-augmented modes accumulate fastest, often invisibly.

But even simple chat degrades. Every exchange adds drag. Conversation length is a poor proxy for context consumption.

Organizations must understand: there is no "safe" mode. There are only modes where degradation is visible versus hidden, fast versus slow, manageable versus opaque.

Organizations deploying AI must understand which modes their workflows use and what coherence limits apply to each.

### What that means for leadership:

- AI behavior you see in short demos does not predict reliability in real workflows.
- "Million-token windows" do not reflect stable reasoning over long contexts.
- High-stakes outputs degrade long before advertised context limits.
- The higher your inputs, the faster degradation will occur.

---

## 3. The Seven Failure Indicators Every Leader Should Know

These indicators form the backbone of the Aggregate Coherence Index (ACI) used in the paper.

They are not technical—they are operational flags:

1. Forgetfulness (context loss, repeated questions)

2. Repetition / Looping
3. Slowdown / Hesitation
4. Instruction-following failure
5. Hallucination (fabricated but fluent information)
6. Self-contradiction
7. Vendor collapse signatures (GPT over-elaboration, Claude verbosity, Gemini contradictions, Grok I/O failures)

Any combination of these signals indicates rising risk.

---

## 4. The Business Meaning of ACI Zones

You don't need to evaluate the AI yourself—revised ACI zones map directly onto organizational risk states.

- Green: Operationally safe
- Yellow: Monitoring required
- Orange: Reliability compromised — human verification needed
- Red: Unsafe for decision-making
- Critical: System nearing failure — immediate reset required

These zones can be integrated into:

- AI governance frameworks
- Risk registers
- Internal audit
- Vendor SLAs

- Enterprise deployment policies
- 

## **5. What Executives Should Require Immediately**

### **A. Vendor Requirements**

Ask vendors to provide:

- Long-context reliability data
- Documented degradation signatures
- Maximum safe interaction lengths (for simple vs complex tasks)
- Multimodal degradation profiles
- Reset/restart recommendations

If a vendor cannot produce these, the model is not enterprise-ready.

---

### **B. Internal Policy Requirements**

Implement:

- Session caps for high-stakes workflows  
(e.g., legal drafting, financial analysis, medical research)
  - Mandatory resets after X exchanges (varies by task)
  - Human verification for outputs in orange/red zones
  - Logging for long interactions in regulated sectors
  - Staff training on degradation detection
-

## C. Deployment Architecture Changes

Shift from free-form conversational usage to bounded, auditable workflows:

- Break long tasks into discrete segments
  - Avoid letting employees use a single conversation as a “project brain”
  - Implement auto-rollover:
    - summarization → new chat → continue
  - Provide domain-specific guardrails
- 

## 6. Executive Summary

The key idea is simple:

AI reliability decays with use.

Vendors don’t disclose it.

Your organization must detect and manage it.

This briefing gives you the operational framework to do so.

---

# Appendix C — Policymaker Briefing

## Long-Context AI Failures: Implications for Safety, Transparency, and Regulation

### 1. The Policy Problem in One Sentence

AI systems degrade invisibly during extended interactions, but vendors provide no disclosures, no warnings, and no reliability guarantees, leaving millions of users in high-risk contexts exposed to silent failure modes.

This represents a clear gap in:

- consumer protection
  - safety governance
  - transparency standards
  - responsible deployment
  - regulatory oversight
- 

## **2. Nature of the Risk**

### **2.1 Silent Degradation**

AI systems do not fail like traditional software. They fail like human cognition:

- Gradual loss of coherence
- Shifts in tone or style
- Confabulatory storytelling
- Forgetfulness
- Contradiction
- Fluently delivered falsehoods

Because output remains confident and natural, users cannot detect failure without training.

### **2.2 Cross-Domain Harm**

Long-context failures create risk across:

- Healthcare
- Law
- Finance
- Education
- Public services
- Critical infrastructure operations

## 2.3 Unequal Burden

Sophisticated users may notice failures.

Vulnerable populations—students, the elderly, the distressed—cannot.

This creates safety inequities.

---

## 3. Evidence Base

The core findings from Evans' Law v5.0 and ACP (AI Conversational Phenomenology):

- Coherence collapses far below advertised context limits
- Collapse follows reproducible scaling behavior
- The same degradation indicators appear across 11+ models
- Multimodal systems fail earlier
- Vendors exhibit distinct failure signatures
- No system provides warnings or self-diagnosis of degradation
- User judgment alone cannot reliably detect the transition from accuracy to coherently wrong output

This makes long-context reliability a regulatory blind spot.

---

## 4. Regulatory Relevance

### 4.1 Not Covered by Existing AI Regulations

Most current frameworks assume:

- benchmark-based evaluation
- discrete, static outputs
- clear system boundaries
- transparent error conditions

Long-context degradation violates all these assumptions.

There are no standards for:

- safe interaction length
- decay curves
- degradation signatures
- disclosure of collapse thresholds
- measurement of progressive failure

### 4.2 Analogy to Known Regulatory Domains

This is similar to:

- drug half-life data (pharmaceuticals)
- fatigue limits (engineering)
- lifetime performance degradation (materials science)
- mean time to failure (hardware reliability)

AI needs use-dependent reliability disclosures.

---

## **5. Recommended Regulatory Requirements**

### **5.1 Mandatory Disclosures**

Vendors should be required to publish:

1. Long-context reliability curves
    - When does coherence begin to degrade?
    - When does collapse become likely?
  2. Known degradation indicators
  3. Vendor-specific collapse signatures
  4. Multimodal degradation rates
  5. Interaction-length safety limits
  6. Recommended reset / restart conditions
- 

### **5.2 Required Testing (Methodological Standards)**

Regulators should adopt or reference ACE/ACP-style testing:

- Structured long-context evaluations
- Defined checkpoints
- Severity ratings for indicators
- Differentiation by drag/complexity
- Text vs multimodal testing

- Reporting of domain-specific safety limits

This allows inter-vendor comparison and independent auditing.

---

### **5.3 Deployment Safety Controls**

Organizations deploying AI in high-risk contexts should be required to:

- Cap max conversation lengths
  - Implement auto-reset after threshold
  - Require human verification in late-stage degradation
  - Maintain logs for long interactions
  - Train staff on degradation detection
  - Use models only within tested reliability envelopes
- 

### **5.4 Consumer Protection Requirements**

Consumer-facing systems (health apps, tutoring apps, financial assistants) must:

- Provide visible warnings during extended interactions
  - Offer built-in session timers or activity meters
  - Display known reliability limits
  - Lock or reset after specified degradation indicators
  - Avoid presenting hallucinated content with high confidence
- 

## **6. Policy Summary**

Long-context failure:

- is predictable
- is measurable
- is universal across vendors
- is undisclosed
- is dangerous
- affects millions of users
- requires explicit regulatory attention

Evans' Law and the ACI/ACP framework provide:

- Testable methodology
- Observed failure indicators
- Practical reliability zones
- A foundation for future regulatory standards

This is a governance gap with immediate safety implications.