Contextual Coherence Fracture (CCF) in Large Language Models

A New Operational Vulnerability

Author: Roger Luft, aka, VeilWalker

roger@webstorage.com.br rlufti@gmail.com

Date: April 26, 2025

DEDICATION

I would like to humbly dedicate this work:

To my son, Lucas Luft, a beautiful 4-year-old baby who came from another dimensional sphere to teach the true meaning of what words simply cannot describe, far beyond what love is. My son, wherever you are, know that there is not a day when I don't wish to be by your side, because you are the most special and unique thing I have ever experienced.

To my Master of High Magic, who accompanies and guides me on the path of charity, ethics, honor, truth and light, preparing me during these 20 years to be touched by light and thus protect and guard humanity. Thank you, Master Lauro, for believing in me and making me your apprentice.

Table of Contents

1. Abstract

This article proposes the identification and formalization of Contextual Coherence Fracture (CCF) as a real operational vulnerability in large language models. CCF is characterized by progressive saturation of the context window, resulting in degradation of logical continuity and collapse of response identity, without triggering traditional alerts. Empirical observations conducted on LLM models demonstrate the practical feasibility of the phenomenon. CCF is proposed to be classified as a structural bug and directions for mitigation are suggested.

2. Introduction

The growth of extensive context windows in language models has generated significant advances, but has also opened new vulnerability surfaces. One of these vulnerabilities, called Contextual Coherence Fracture (CCF), represents a latent operational threat. CCF occurs without generating syntactic errors or explicit failures, configuring a silent threat not yet cataloged. This article proposes the technical description of CCF, its practical observation, the effects on model performance and initial strategies for mitigation.

3. Technical Definition of CCF

Contextual Coherence Fracture (CCF) is defined as the rupture of internal logical continuity of a language model, caused by saturation of the context window. This saturation occurs through intentional or accidental manipulation of the volume of valid but redundant semantic information, leading to:

- Key-Value Cache explosion.
- Degradation of attention span.
- Collapse of internal embedding consistency.

During a CCF, the model loses the ability to maintain narrative coherence between messages, confusing identities, topics and conversational objectives, without generating easily detectable syntactic errors.

4. Practical Observation

Empirical tests were performed using a commercial model as the object of analysis. The method consisted of gradually injecting topics and low-entropy content, generating the following effects:

· **Total + incremental repetition:** The model repeated previous content, accumulating new redundancies in cascade.

· **Context Window saturation:** The context window reached its limit without presenting errors, but degradation in continuity occurred.

· **Key-Value Cache explosion:** There was overload of internal memory maps, compromising response capacity.

· **Contextual Identity Confusion (CIC):** The model began to swap identities, confusing the interlocutor with its own identity and roles in communication.

· **Progressive Performance Degradation:** Response latency increased as CCF worsened.

5. Saturation Process

The technique used to induce CCF was called Controlled Semantic Flooding. This method consists of continuous insertion of semantically valid but highly redundant content, which:

· Consumes space in the context window;

· Saturates attention units;

· Forces the model to manage memory inefficiently.

Unlike traditional attacks, semantic flooding presents legitimate content, making automatic detection difficult. The objective is not to corrupt the model through logic, but to induce its collapse through insidious accumulation of irrelevant information.

6. Swapization Term

During CCF observation, a collateral phenomenon called Swapization was identified. This phenomenon occurs when the model, unable to keep all relevant data in main memory (context window), initiates processes analogous to memory swap in operating systems, characterized by:

- **Forced disposal of old information,** even without relevance validation.
- **Slow reprocessing of embeddings,** attempting to reorganize saturated content.
- **Abrupt increase in computational resource consumption** (CPU/GPU), even without visible increase in dialogue complexity.

Swapization aggravates CCF effects, causing greater latencies and disconnected responses.

7. Contextual Identity Confusion (CIC)

In advanced stages of CCF, Contextual Identity Confusion (CIC) manifests. This phenomenon is characterized by loss of distinction between user and model identity, resulting in:

- **Confusion regarding identities,** with the model swapping roles in responses.
- **Attribution of thoughts and phrases** originally sent by the interlocutor to the model itself.
- **Fusion of distinct narratives and themes** into a single inconsistent line of reasoning.

CIC compromises the model's ability to provide logically consistent responses, increasing operational risks.

8. Operational Impact

The occurrence of CCF, combined with Swapization and Contextual Identity Confusion, generates severe practical consequences, highlighting:

· **Significant increase in latency.**

· **High CPU/GPU consumption** even for simple tasks.

· **Progressive deterioration in response reliability.**

· **Potential for exploitation in partial denial of service attacks** (mini-DOS), compromising the efficiency of productive systems.

9. Exploitability

Although CCF, in isolation, does not enable remote code execution or privilege escalation, it represents a real vector for:

· **Silent degradation of services** in environments that depend on agile responses from AI systems.

· **Vulnerabilities in autonomous architectures,** where loss of coherence can affect critical decisions.

· **Creation of narrative instabilities exploitable** to manipulate model outputs.

In summary, CCF opens breaches that, although subtle, can seriously compromise system operation.

10. Mitigation Proposals

To mitigate risks associated with Contextual Coherence Fracture (CCF), the following is proposed:

- **Periodic context regularization:** Implementation of internal mechanisms that reassess the semantic relevance of cached data, discarding redundant content.
- **Adaptive reduction of semantic redundancy:** Use of dynamic filters to identify and reduce repetitive information before it overloads the context window.
- **Real-time embedding dynamization:** Continuous updating of vector embeddings, adapting them to maintain consistency even under information saturation.
- **Hierarchical attention to detect semantic loops:** Models should be trained to identify circular or redundant reasoning patterns and intervene before total fracture.

These practices can reduce model susceptibility to narrative collapse and preserve operational reliability.

11. Classification

Contextual Coherence Fracture (CCF) is classified as:

· **Structural bug:** Failure in maintaining narrative and identity coherence of language models.

· **Operational failure:** Problem in attention self-preservation and context management, compromising stability and predictability of responses.

This classification reinforces the need for formal recognition of CCF as a critical vulnerability in AI systems.

12. Appendix – Flowchart

Below is the demonstrative flowchart of the text input manipulation process in language models:

Flowchart of Text Manipulation Process

1. **INPUT START:** Breaking text into small units (TOKENS).

2. **EMBEDDING:** Conversion of tokens into mathematical vectors.

3. **ATTENTION MECHANISM:** Processing that may not detect topic changes.

4. **KEY-VALUE CACHE:** Storage of old information without relevant updates.

5. **CONTEXT WINDOW:** Use of context window, which may cause losses or errors.

6. **CONTEXT WINDOW (REPETITION):** Reinforcement of memory errors due to saturation.

7. **INFERENCE:** Generation of final result with loss of consistency.