TECHNICAL BRIEF

Context Degradation in Long-Form AI Conversations

The case for structured context curation and the FUR / Historian system

Prepared by: Andrew R. Garcia, Ph.D.
Role: Independent AI Systems Consultant
Date: April 2026
Version: 1.0 — Client Distribution

Executive Summary

Large language models degrade in measurable, predictable ways as conversations grow longer. This is not a minor inconvenience. It is a structural property of how these systems process context, and it has direct consequences for any professional use case that depends on multi-turn AI reasoning: legal analysis, financial modelling, policy development, and complex technical workflows.

This brief explains the problem, grounds it in peer-reviewed research, and introduces a practical solution: the **FUR engine**, implemented as **Rust Logic**, a CLI tool, and **Historian**, a GUI layer that makes context curation accessible to non-technical users. Together, these tools allow practitioners to actively manage what an AI system remembers across a long conversation, reducing drift, contradiction, and unsupported inference.

The mechanism is grounded in established research on context degradation in long-form AI interactions, and the system is designed to directly address these failure modes in practice. The problem is well-documented in the literature and increasingly visible in professional use, and this approach provides direct control over conversational context and continuity.

The Problem: What Happens to AI Reasoning Over Time

When a professional uses an AI system for a sustained reasoning task, such as drafting a contract, working through a regulatory question, or debugging a complex model, the conversation typically grows to tens or hundreds of turns. This is where things go wrong.

Attention dilution and positional bias

Language models do not read a conversation the way a human reads a document. They assign attention across the entire context window, and that attention is not uniformly distributed. Research from Stanford, published in 2023, demonstrated that models systematically underperform when critical information appears in the middle of a long context, even when that information is directly relevant to the query. Performance was strongest when relevant content appeared at the beginning or end, a pattern the authors termed the “lost in the middle” effect.

Liu et al., 2023

“Lost in the Middle: How Language Models Use Long Contexts” — Stanford University. Demonstrated consistent performance degradation on multi-document QA and key-value retrieval tasks when relevant content was positioned centrally in long contexts.

Irrelevant context as active interference

A common assumption is that additional context is neutral at worst, meaning that including irrelevant turns simply adds noise but does not actively harm reasoning. This assumption is wrong. Work by Shi and colleagues showed that semantically irrelevant information introduced into a context window does not merely fail to help: it actively degrades model performance on reasoning tasks, including tasks where the model would otherwise perform well.

This has a direct implication for professional use. Every exploratory turn, every digression, and every clarifying question that is answered and then abandoned is not inert. Each competes for attention with the constraints, facts, and commitments that actually matter to the task.

Shi et al., 2023

"Large Language Models Can Be Easily Distracted by Irrelevant Context" — Google DeepMind. Showed that irrelevant sentences added to reasoning problems caused significant accuracy drops across multiple LLM families.

Position drift in multi-turn reasoning

Perhaps the most professionally consequential failure mode is **position drift**: the tendency of a model to gradually shift its stated conclusions, abandon earlier commitments, or contradict prior reasoning as a conversation extends. This is not hallucination in the conventional sense. The model is not fabricating facts. It is losing track of its own prior reasoning chain.

In a legal or financial context, this means a model that correctly identified a constraint at turn 5 may reason as though that constraint does not exist at turn 25. The output looks fluent and confident. The error is invisible unless the practitioner explicitly checks back.

Why Existing Solutions Are Incomplete

Several approaches already exist to manage context in AI systems. None of them address the specific problem that arises in live, professional, multi-turn conversations.

Retrieval-Augmented Generation (RAG)

RAG systems retrieve relevant documents from a knowledge base and inject them into the context at query time. This works well for static document retrieval but does not address conversations that generate relevant context dynamically. Key constraints may emerge in turn 7 and must persist to turn 30. RAG assumes that important content already exists in a retrievable form. In live reasoning conversations, this assumption does not hold.

Automatic summarisation

Some systems automatically compress older conversation turns into summaries to manage context length. The problem is well-documented in the summarisation literature: compression loses logical commitments. A summary that says "the user asked about liability" does not preserve the specific position the model staked out on that question. When that position needs to be reconciled with new information later, it is gone.

Simple truncation

The most common approach in practice is the simplest: drop the oldest turns when the context window fills. This is more damaging than it appears. Early turns often contain the foundational constraints of the task, including the problem statement, key facts, and agreed scope. Removing them removes the anchor.

The gap

No existing production solution gives a practitioner direct, selective control over which turns persist across a long conversation while preserving the reasoning integrity of those turns. This is the problem FUR and Historian are designed to solve.

The Approach: FUR, Rust Logic, and Historian

The **FUR engine** (Focused Understanding and Retention) is the core mechanism. It treats a conversation not as a flat chronological log but as a structured set of turns with varying epistemic weight. Some turns establish constraints, some explore possibilities, and some are noise. FUR allows a practitioner to select which turns are retained and re-injected into the active context as a conversation grows.

Rust Logic

Rust Logic is the CLI implementation of the FUR engine, built for technical users and pipeline integration. It operates directly on conversation logs, applies the curation logic, and outputs a structured context file ready for re-injection. For developers and AI engineers integrating this into existing workflows, Rust Logic is the entry point.

Historian

Historian is the operational interface for institutional use, built on top of the FUR engine. It is designed for professional users who are not developers, including lawyers, analysts, policy researchers, and consultants, who need the benefits of context curation without working at the command line. Historian provides structured interaction with conversation data, enabling turn-level selection, context control, and consistent reuse across sessions.

This separation is deliberate. Rust Logic addresses the technical market. Historian extends the addressable client base to institutions whose staff use AI for sustained professional reasoning tasks.

What this achieves mechanically

- **Retention:** Key constraints established early in a conversation survive to later turns, rather than being diluted or truncated away.
- **Noise reduction:** The active context presented to the model at any point contains only turns the practitioner has judged relevant, reducing the interference effect documented by Shi et al.
- **Auditability:** Because the practitioner decides what persists, they also review what they retain. This constitutes a lightweight audit of their own reasoning process.
- **Model-agnostic:** The approach works with any underlying model. It is infrastructure, not a model replacement.

What the Literature Predicts We Should See

Formal evaluation is underway, focusing on measurable differences between curated and uncurated conversational contexts across complex reasoning tasks. The expected effects follow directly from established findings on attention distribution, context interference, and long-context degradation.

The FActScore framework (Min et al., 2023) provides a methodology for measuring factual precision in long-form generation by decomposing outputs into atomic claims and verifying each. Applied to curated versus uncurated conversation outputs, this framework would predict higher claim-support rates in curated conditions. The difference arises from the greater coherence of the context the model is reasoning from.

Min et al., 2023

"FActScore: Fine-grained Atomic Evaluation of Factual Precision in Long Form Text Generation." Provides a replicable framework for measuring the proportion of claims in a generated output that are supported by a given source context.

The lost-in-the-middle findings predict that models reasoning from a curated context, where the most relevant turns are explicitly present and not buried, should show more consistent performance across the length of a conversation. Specifically, we would expect fewer position reversals and fewer instances of a model ignoring a constraint it had previously acknowledged.

These are directly testable effects, and the evaluation design focuses on measuring them in realistic professional workflows.

Who This Is For

The context degradation problem is most acute wherever AI is used for sustained, complex, multi-turn reasoning. This applies not to one-shot queries but to extended analytical conversations where the output at turn 40 must remain consistent with the constraints established at turn 5.

- **Legal institutions:** Contract analysis, regulatory research, and any legal reasoning task where constraint tracking is essential and errors are consequential.
- **Financial institutions:** Risk modelling workflows, scenario analysis, and compliance reasoning that extends across multiple sessions or user-AI exchanges.
- **Research and policy bodies:** Internal AI deployments where staff are using models for policy development, research synthesis, or strategic analysis.
- **Technical organisations:** Engineering teams using AI for architecture decisions, debugging, or technical specification, where a wrong turn at step 30 can be traced to a forgotten constraint from step 3.

Next Steps

This brief is the opening of a conversation, not the close of one. The right next step depends on where you are.

- If you are a **technical team** evaluating integration possibilities, Rust Logic is available for review. We can walk through the engine design, the curation logic, and how it would sit within your existing infrastructure.
- If you are an **institutional user** whose staff are working with AI on extended reasoning tasks, Historian is the relevant entry point. We can arrange a demonstration and discuss deployment.
- If you are a **research body** interested in participating in the formal evaluation, for example by contributing domain-specific conversation data or co-designing evaluation protocols, we are actively looking for institutional partners.

Andrew R. Garcia, Ph.D. — garcia.gtr@gmail.com

Key References

Liu et al.	Liu, N. F., Lin, K., Hewitt, J., Paranjape, A., Manning, C. D., & Liang, P. (2023). *Lost in the Middle: How Language Models Use Long Contexts.* arXiv:2307.03172.
Shi et al.	Shi, F., Chen, X., Misra, K., Scales, N., Dohan, D., Chi, E., et al. (2023). *Large Language Models Can Be Easily Distracted by Irrelevant Context.* arXiv:2302.00093.
Min et al.	Min, S., Krishna, K., Lyu, X., Lewis, M., Yih, W., Koh, P. W., et al. (2023). *FActScore: Fine-grained Atomic Evaluation of Factual Precision in Long Form Text Generation.* arXiv:2305.14251.
Shi et al.	Shi, W., Min, S., Yasunaga, M., Seo, M., James, R., Lewis, M., et al. (2023). *REPLUG: Retrieval-Augmented Black-Box Language Models.* arXiv:2301.12652.