PANTHEON

RESEARCH PAPER / AUGUST 2026

Context continuity: why capable agents still lose the thread

Long context is not long-term memory, and a summary is not the record. A research-backed account of the continuity problem and Pantheon's architecture for preserving the thread across sessions, models, and applications.

Archive first. Distil second. Retrieve on demand.

pantheon-ui-web.vercel.app/context-continuity

ABSTRACT

The problem is not intelligence. It is state.

A language model can reason capably inside one interaction and still lose the work when the session, context window, application, or model changes. We call the missing system property context continuity: the ability to preserve, carry forward, and selectively reconstruct the state needed to continue work across those boundaries.

This is a working product and systems definition, not yet a settled academic category with one standard benchmark. It is useful because it separates continuity from three narrower ideas: the tokens currently inside a context window, a retrieval store that can return similar text, and a summary of an earlier interaction. Each can help. None guarantees continuity alone.

> **Continuity does not mean replaying everything. It means preserving enough evidence that the system can assemble the smallest trustworthy context for the work in front of it.**

01

Why context breaks

The obvious response to agent forgetting is a larger context window. Research shows why that answer is incomplete. Lost in the Middle found that model performance can change sharply with the position of relevant information. Models often performed best when the needed fact appeared near the beginning or end of an input and worse when it appeared in the middle, even when the models were designed for long context [1].

RULER broadened the test beyond retrieving one hidden fact. It added multiple needles, multi-hop tracing, aggregation, and question answering. Almost all of the 17 tested models degraded as input length and task complexity increased. Although every model claimed a window of at least 32K tokens, only half maintained the benchmark's satisfactory level at 32K [2]. A nominal context window is capacity, not a guarantee that every token will influence an answer equally.

LongMemEval reaches the same problem across time instead of inside one prompt. Its 500 questions evaluate information extraction, multi-session reasoning, temporal reasoning, knowledge updates, and abstention over sustained interaction histories. The authors report a 30 percent accuracy drop for tested commercial and long-context systems on long-term memory tasks [3].

Middle Relevant information can become harder to use when it sits between the beginning and end of a long prompt [1].

Half Only half of the 17 RULER models maintained satisfactory performance at 32K tokens [2].

30% LongMemEval reports an accuracy drop across sustained interactions for tested systems [3].

02

Four discontinuities

- **The session boundary**
  A model call has no inherent memory of an earlier call. A new chat begins from the context supplied now, not from a durable understanding of the project.
- **The attention boundary**
  A larger window can hold more tokens, but holding information is not the same as using it reliably. Position, distractors, and reasoning complexity all affect recovery.
- **The compression boundary**
  Compaction keeps work moving, but every summary makes a selection. A detail discarded today may be the evidence required to diagnose tomorrow's failure.
- **The application boundary**
  Conversation state normally belongs to the client that recorded it. Switching agent applications often loses the thread even when both use capable models.

  Compaction remains valuable. Anthropic describes it as a first lever for long-horizon coherence, while warning that aggressive compression can remove subtle details whose importance only becomes clear later [6]. The design implication is not to avoid summaries. It is to avoid making the summary the only surviving account.

03

What continuity requires

Earlier memory architectures point toward a hierarchy rather than one giant prompt. MemGPT treats active context like RAM backed by external storage that can be paged in through tools [4]. Generative Agents keeps a comprehensive memory stream, synthesizes higher-level reflections, and retrieves memories according to relevance, recency, and importance [5]. Anthropic's context-

engineering guidance states the same resource constraint directly: useful context is the smallest set of high-signal tokens that improves the desired outcome [6].

- **Durable evidence**
  Preserve the source conversation and produced files outside the model window. Distillation should sit above the record rather than replace it.
- **Layered representation**
  Treat recent orientation, durable facts, episodic history, artifacts, and procedural guidance as different layers with different retrieval costs and lifetimes.
- **Progressive disclosure**
  Start with a bounded map, then retrieve a matching session, turn range, or file only when the task needs it. Continuity should conserve attention rather than fill it.
- **Provenance and change**
  Retain stable session identities, timestamps, source turns, and version history so conclusions remain traceable and later updates do not silently erase earlier state.
- **User control**
  Keep the durable record in storage the user can inspect, export, edit, or delete. Portability is incomplete when memory is locked inside one model vendor.
- **A common access protocol**
  Give different agents the same verbs: orient, search, open a session, fetch a file, and load relevant guidance. An open protocol makes that vocabulary portable.

04

Pantheon's design

Pantheon is a continuity layer for desktop AI agents, not a model and not another chat client. A local Model Context Protocol process reads the session record the agent application already writes, then stores a named project conscience in a private GitHub repository owned by the user. MCP gives different hosts a common tool surface for handing off, resuming, searching, and opening evidence [7].

The central design decision is archive first, distil second. The packet that orients the next agent can remain short because the underlying sessions and files survive outside it. If the packet is insufficient, the agent has stable ways to inspect what it summarized.

- **Orientation - handoff.md**
  A bounded space-level packet rebuilt from the archive: current state, earlier packet summaries, recent spoken turns, totals, session references, and stored file names.

- **Durable knowledge - memory.md**
  Facts and decisions that should outlive one episode. They append across handoffs instead of being regenerated from the newest chat.

- **Episodic record - sessions/<id>/transcript.jsonl**
  The archived conversation in addressable turns, including the tool activity the client actually records. Each session stays independently readable.

- **Historical interpretation - handoffs/<time>.md**
  The packet written at each handoff. Earlier interpretations remain visible, so a short final session cannot replace the project's previous account of itself.

- **Artifacts - sessions/<id>/artifacts/**
  Generated or attached files stored as bytes, not merely mentioned in prose. Resume lists them and fetches contents only on demand.

- **Procedural memory - skills/**
  Reviewed lessons with evidence and triggers. Resume advertises promoted skill names and triggers; the full instruction is loaded only when the task matches.

  The representation is deliberately file-shaped. Plain text gives humans and agents the same inspectable substrate. Git adds version history and transport without making Pantheon's application database the owner of the conversation. Conversation content stays in the user's repository; the web application stores account, pairing, sharing, and token metadata.

  The design is also model-neutral. A conscience is not serialized into one vendor's private memory API. Any connected MCP client can receive the same orientation and call the same retrieval tools, subject to the capabilities of that host.

05

From handoff to retrieval

- **1. Capture the episode**
  Pantheon identifies the active local session, archives new turns without duplicating the stored prefix, collects eligible artifacts, and keeps that episode under a stable session id.

- **2. Build a bounded orientation**
  It writes a session packet and rebuilds the space-level handoff from all sessions: totals, recent spoken turns, earlier packet summaries, session references, and artifact names.

- **3. Version the state**
  The changed files are committed and pushed to the user's private vault. Later handoffs append another episode or tail rather than flattening the project into one mutable recap.

- **4. Resume with a map**
  The receiving agent gets the packet, durable memory, session identities and sizes, artifact names, and promoted skill triggers. Orientation does not consume the window with the whole archive.

- **5. Retrieve the evidence**
  When a question exceeds the packet, the agent searches messages and recorded tool calls, opens surrounding turns or a full session, and fetches named artifacts.

> **The packet is not the memory. It is the index card attached to the memory. The transcript, decisions, files, and version history are what make its claims reopenable.**

06

Limits and open work

Context continuity is infrastructure, not a guarantee of correct reasoning. A receiving model can ignore a clue, search poorly, or apply an old decision to a changed situation. Pantheon improves what can be carried forward and inspected; it does not make the model infallible.

- **Capture is limited by the host record**
  Pantheon can archive only what the desktop client writes locally. Some clients keep full turns; others omit tool results or expose no readable session. Missing evidence is reported rather than invented.

- **Browser-only agents lack full-fidelity capture**
  A remote MCP server cannot read a browser client's private conversation log. Pantheon therefore focuses setup on desktop agents that can start a local process.

- **Search is currently lexical**
  Exact phrases, commands, file names, and tool calls are strong keys. Paraphrased or conceptually related memories may require broader inspection. Future semantic indexing must preserve stable addresses and provenance.

- **Distillation remains a judgement**
  A packet and a proposed skill are model interpretations. The raw episode remains available beneath the packet, while skills stay inert drafts until a human promotes them.

- **A Git vault is controlled, not magical**
  Users can rewrite or delete their repository, and security still depends on GitHub and machine credentials. Ownership makes those controls explicit; it does not remove the need to secure them.

  The next useful evaluation target is the handoff boundary: can a different agent recover the correct decision, evidence, artifact, and unresolved work after multiple sessions? That is a stronger test than asking whether one stored sentence can be retrieved. LongMemEval's separation of extraction, temporal reasoning, updates, and abstention offers a useful starting vocabulary [3].

07

Conclusion

Models will gain larger windows, and compaction will improve. Neither removes the architectural need to decide what persists, where it lives, how it changes, and how another agent can recover evidence without ingesting an entire project history.

Pantheon's answer is concrete: a loss-minimizing archive, a compact orientation layer, granular retrieval, model-neutral MCP tools, and a private Git repository the user owns. Preserve the thread, disclose it progressively, and make every summary reopenable.

SOURCES

References

Research was reviewed from primary papers and official protocol or engineering documentation. Claims about Pantheon describe the current implementation rather than results from an independent benchmark.

[1] Lost in the Middle: How Language Models Use Long Contexts
Liu et al., 2023
https://arxiv.org/abs/2307.03172

[2] RULER: What's the Real Context Size of Your Long-Context Language Models?
Hsieh et al., COLM 2024
https://arxiv.org/abs/2404.06654

[3] LongMemEval: Benchmarking Chat Assistants on Long-Term Interactive Memory
Wu et al., ICLR 2025
https://arxiv.org/abs/2410.10813

[4] MemGPT: Towards LLMs as Operating Systems
Packer et al., 2023
https://arxiv.org/abs/2310.08560

[5] Generative Agents: Interactive Simulacra of Human Behavior
Park et al., UIST 2023
https://doi.org/10.1145/3586183.3606763

[6] Effective context engineering for AI agents
Anthropic Applied AI, 2025
https://www.anthropic.com/engineering/effective-context-engineering-for-ai-agents

[7] Model Context Protocol specification
Model Context Protocol
https://modelcontextprotocol.io/specification/latest