Compaction and the Evidentiary Record

Flyxion
Independent Researcher

November 2026

Abstract

A recent line of work on agents with deliberately small working contexts reports that a model can often continue a substantial task from a compact account of its own progress, provided that the documents, files, and environment it acts on remain accessible. This essay develops a distinction acknowledged in that work: reducing what occupies the model's immediate attention differs from eliminating the evidence on which its conclusions depend. The reported experiments establish the practical value of the former while leaving open how reliably compacted agents recover and apply evidence under changed questions. The benchmarks ask whether enough survived to obtain the scored result, whereas a stronger criterion asks whether enough survived to support warranted continuation, including changed questions, contested conclusions, and later demands for the evidence behind an earlier decision. The essay argues that a handoff summary is a derived working state and not a verified projection of the record, identifies the latency of evidence as the characteristic failure of compact states backed by archives, and sketches checks that would make the preservation criterion measurable. The essay implements none of these checks, and it claims only that the empirical results leave the distinction open.

1 Two Ways a Context Becomes Large

A context grows for one of two reasons, and the remedies differ accordingly. In a long-horizon task, such as repairing a software repository, the conversation itself accumulates commands, observations, abandoned approaches, and intermediate decisions, so that the context at step two hundred is mostly the residue of the preceding steps. In a long-input task, such as answering a question that depends on a large collection of documents, the conversation may be short but the supplied material is not, and the context is large because the input was handed over whole. The paper discussed here treats both under a single proposal: keep the working context small by summarizing accumulated progress when necessary, and by retrieving relevant passages from external files in place of loading all available material into every inference [20].

The proposal is easy to state and easy to misread. A reader who takes it to say that a model needs less information than was previously supposed will conclude something the experiments

do not show. What the experiments show is that a model needs less information *present at once*, so long as the remainder is reachable. That is a claim about the arrangement of information across time and storage, and it leaves open whether the arrangement is adequate for tasks other than the ones scored.

The separation between immediate context and persistent resources has an architectural precedent in MemGPT, which manages information across memory tiers to support tasks exceeding the model's immediate context capacity [17]. Such architectures relocate the problem of continuity: information must not only remain stored, but must be selected and returned to the working context when required.

2 What the Experiments Establish

The paper compares three operating regimes. The baseline preserves the entire conversation. A second regime periodically replaces the conversation with a handoff summary while retaining the original task statement and the persistent environment. A third and more radical regime performs one environment action, observes its result, writes an updated summary, and begins a fresh conversation for the next action. The contexts called short are generally between about sixteen and fifty thousand tokens, so the smallness in question is relative and not extreme [20].

On the long-horizon benchmarks, periodic compaction usually keeps performance within a few percentage points of the full-history baseline and sometimes exceeds it. On SWE-bench Verified [5, 8] the best tested compaction setting raises one of the two model families studied from 70.0 to 74.8 percent and the other from 84.4 to 88.8 percent. These are best settings among those tried, and the reasonable inference is the weaker one: substantial reductions in working context need not produce substantial deterioration in task completion. The single-action regime is more surprising. On SWE-bench Verified one model family scores 70.2 percent against its 70.0 percent baseline while the other scores 82.4 against 84.4. On Terminal-Bench 2.1 [7][1] the first improves from 47.2 to 51.8 percent while the second falls from 87.6 to 84.3. Median peak contexts under this regime are only about seven to nine thousand tokens. Useful continuity therefore survives repeated conversational resets, though this does not establish that resetting after every action is the best policy in general [20]. A separate qualification concerns benchmark validity. Before the paper's publication, OpenAI reported flawed tests in an audited subset of SWE-bench Verified and evidence of training contamination, and announced that it had stopped reporting scores on the benchmark [9]. These concerns limit the interpretation of absolute scores as measures of general software-engineering capability. They do not by themselves invalidate comparisons between context-management regimes using the same model, although the extent to which contamination affects those comparisons remains unmeasured.

[1]The paper discussed here evaluates version 2.1 of the benchmark and cites the original benchmark paper, which presents version 2.0. The figures reported here are for version 2.1.

On the long-input side, the largest gains come from selective reading and not from compaction. The agent is given filenames and uses shell tools to search and inspect the documents, and on several benchmarks it finishes without compacting at all. The configuration reaches complete accuracy on a needle-in-a-haystack evaluation [6], and BABILong [1] accuracy rises from 67.1 to 92.0 percent for one model family and from 70.7 to 94.0 percent for the other [20]. The comparison here is between a model with searchable external documents and a model that ingests the complete input, and these are different computational arrangements, not the same computation made cheaper.

The economic claims require a qualification that the headline numbers invite readers to skip. The measured serving experiments on open models find that generating tokens at a context of two hundred fifty-six thousand tokens costs roughly 3.6 to 7 times as much as at sixteen thousand, under the tested hardware and serving conditions. The benchmark tables also report estimated optimized retail costs derived from a context-dependent cost model, and these should not be read as demonstrated reductions in actual billing for closed models. The LongBench v2 [2] table makes the point itself, since the nominal retail cost for one model family rises from $79.20 to $90.20 under the short-context configuration even as the modeled optimized cost falls sharply [20]. The anomaly matters because it shows a configuration that is cheaper under a model of how serving cost scales with context and more expensive at the nominal prices a user of a closed model would face, so the two cost claims cannot be merged into one. The cost argument is therefore sound as a statement about serving at long context and unproven as a statement about what a user of a closed model will pay.

Finally, the paper proposes what it calls a semantic state space model: a model trained to receive an explicit linguistic description of the current state, act, observe, and rewrite the description so that another inference instance can continue. The experiments use existing models inside a harness. The benefits of training a model specifically for this pattern remain a proposal, and nothing in the reported results measures them.

3 Working State and Evidentiary Record

The central distinction of this essay can be stated as a pair of definitions.

Definition 1 (Working state)**.** *A working state is a bounded representation, held in the model's immediate context, that is sufficient to select the next action of a task under the questions the task is expected to pose.*

Definition 2 (Evidentiary record)**.** *An evidentiary record is an append-only account of recorded actions, observations, and outcomes against which later historical claims can be checked, subject to the record's coverage and the reliability of its sources.*

The two differ in what they are required to do. A working state is judged by whether

it supports good next actions, and it benefits from being small, selective, and revisable. An evidentiary record is judged by whether it can answer questions that were not anticipated when it was written, and it benefits from being complete and unrevised. A bounded, lossy working state cannot generally substitute for an evidentiary record across unrestricted future questions about an accumulating history, because compression discards and a representation that has discarded something can no longer adjudicate questions about it. Its adequacy is relative to the distinctions it preserves and the continuations it must support. A representation can serve both roles in restricted cases: a short history may fit entirely within a working state, a lossless encoding may preserve it, and a structured object may hold both a summary and an immutable record. Success as a working state does not, however, establish adequacy as a record.

The paper acknowledges this directly. In its third section it imagines a tool that emits successive coin tosses. A bounded summary must eventually lose some outcomes, which makes an arbitrary later query about the sequence impossible to answer reliably from the summary alone. The paper's resolution is to preserve the complete conversation externally and to give the model selective access to it [20]. This is the correct resolution, and it already contains the distinction drawn above: the compact state guides the next operation, the archive supplies what may later need to be recovered, and the empirical results concern how little must remain immediately present. The paper's contribution is thus best read as a result about working states backed by persistent records, and it is weaker as evidence that working states can stand alone.

An append-only history paired with a derived working state has a familiar architectural counterpart in event sourcing, where changes to application state are retained as a sequence of events and state can be reconstructed from that sequence [12]. The analogy concerns the relation between a retained history and its derived representation, and not any guarantee that the recorded events are complete or correct. In a ledger-based system the current balance is derived from the transaction history and is trusted because the derivation is deterministic and can be rerun. The database literature on provenance distinguishes why, how, and where provenance for derived data [3], and tamper-evident logging obtains its guarantees from cryptographic structure and not from readability [10, 4]. The derived value is useful precisely because it is disposable: if it is lost or doubted, it is recomputed from the record. The natural-language handoff summary lacks a specified preservation relation whose satisfaction the harness checks against the record. Rerunning the summarizer may produce different wording, but textual variation is not itself the problem: different summaries can preserve the same relevant distinctions, and a deterministic summary can omit them consistently. The question is whether the derived state preserves the claims, qualifications, and evidentiary relations required for its intended use. Readability is not verification. A summary can read as a faithful account while omitting an exception, promoting a conjecture to an established fact, or preserving a conclusion while losing the observation that qualified it. Research on factual consistency in abstractive summarization has accordingly

treated agreement with the source as a separate evaluation target, including the identification of supporting source spans and inconsistent summary spans [14]. Applying such checks to agent handoffs would require extending them to actions, verification status, qualifications, and unresolved constraints.

The same structure is familiar from scholarly writing, which suggests that compaction is a normal feature of how knowledge is produced and not a peculiarity of language-model agents. An essay is a working state derived from a much larger record of reading, skimmed material, lectures, and working notes, and that record is often longer than the essay and is rarely published with it. A reader who receives only a further compression, such as a spoken or written overview of the essay, holds a summary of a summary, usually without access to the notes behind either. At each step qualifications can be dropped and the status of claims can change, so the question of which statements remain operative arises long before any agent harness is involved. What distinguishes a well-run process is not the absence of compression but the survival of a record behind each summary, together with some means of consulting it when a claim is challenged.

4 What the Compaction Prompts Do and Do Not Guarantee

The compaction prompts used in the paper are more disciplined than generic summarization. They instruct the model to separate observations from unverified ideas, to preserve exact paths and source locations, to record verification outcomes, and to retain failed approaches and unresolved work [20]. Each of these instructions targets a known way in which summaries degrade, and their presence suggests the authors anticipated the concerns raised here.

An instruction to preserve verification status is nevertheless a different thing from verification that the status was preserved correctly. The prompt specifies what a good summary contains; it does not check that the summary produced contains it. The gap between specification and check is the ordinary gap between a requirement and a test, and it matters here because summarization errors need not cancel across successive compactions; some may persist or reinforce one another. One plausible failure mode is an upward drift in apparent certainty, as conclusions survive while their qualifications disappear. A summarizer under pressure to be brief may prefer the assertion to its qualification, the conclusion to its derivation, and the settled to the pending, because the former are shorter. The paper does not establish this tendency, and compaction can equally remove overconfident speculation, so the direction of drift is a hypothesis to be measured. Repeated compaction could amplify such a failure, and whether it does is likewise a quantity the proposed evaluation should record and not assume.

The point that follows is about the insufficiency of an evaluation criterion, and it can be stated as a proposition.

Proposition 1. *Let two working states support the same scored outcome on a fixed objective,*

and let both have access to the same complete evidentiary record. They can still differ in their behavior on subsequent questions about the task's history, because access to evidence does not determine whether it is recognized as relevant, retrieved, and applied correctly. Success on the scored objective therefore does not, by itself, establish preservation of the evidence and qualifications required for other continuations.

The claim concerns the insufficiency of the evaluation criterion and not a necessary decline under repeated compaction. A scored objective may depend on only a small part of the record, leaving differences in the treatment of other evidence undetected. Archive access preserves the availability of that evidence, but it does not guarantee its retrieval or correct application. Completion rate therefore supplies evidence about the evaluated objective without, by itself, establishing preservation under changed questions. What may deteriorate under compaction is not the information available in the working state together with the record, which has not necessarily diminished, but the agent's ability to recognize which evidence matters, retrieve it, and use it correctly within a practical budget.

5 The Latency of Evidence

Moving the evidence into an archive removes the risk of permanent deletion and introduces a different risk. A fact may remain fully available in storage while disappearing from the agent's immediate account of the situation, and recoverability then depends on whether the agent has a reason to search for it. Availability and effective use are already distinguishable within the context window itself. Liu et al. find that performance can depend strongly on where relevant information appears, with evidence in the middle of long contexts often used less successfully than evidence near its boundaries [15]. The position effect shows that availability within the context is already distinct from effective use, and the archive case extends this distinction across the context boundary. The mechanism differs, since position effects concern evidence already present, but the implication is parallel: archived evidence may never enter the working context because the agent does not initiate the relevant retrieval. Retrieval may be prompted by the working state, a new question, standing instructions, or an independent audit procedure. When none of these supplies the missing cue, an omitted qualification can remain unconsulted despite being fully recoverable. An omitted qualification is especially troublesome because its absence removes the cue that the working state would otherwise have supplied. If a summary states that a fix works while omitting that it was verified only on one platform, the agent may generalize the result without checking the archived qualification. Editorial collaboration supplies another example. A handoff may retain a proposed correction while omitting that the author subsequently rejected it, or preserve an earlier draft's objection after the latest revision has resolved it. The resulting error concerns the status of a statement within the history: remembering that a suggestion occurred does not establish that it remains applicable. A useful

working state must therefore preserve supersession and resolution, as well as the content of earlier claims, and this is one reason an append-only record and a revisable working state complement one another. A related failure appears in human-factors research on automation bias, where participants sometimes failed to respond to events that an automated aid did not explicitly indicate [18]. This provides an analogy for dependence on supplied cues, and not direct evidence that compacted agents exhibit the same mechanism.

This failure is distinct from forgetting and from hallucination, and it is not necessarily resolved by improving archive completeness or retrieval capability alone. It is a failure of noticing, in which the information exists and the access path exists and the trigger for using them does not. The selective-reading results on long-input tasks show that agents search competently when the question tells them what to look for. The harder case is the one in which the working state must tell the agent that there is something to look for, and the state has been compressed by a process that cannot know which absences will later matter.

External preservation thus solves one problem and exposes another. What it solves is evidence loss. What it exposes is the dependence of retrieval on the content of the very state that compaction has shortened.

6 Making the Preservation Criterion Measurable

The preservation criterion can be turned into procedures, and doing so is where a verification framework adds something the benchmarks do not supply. The ledger comparison of Section 3 supplies the organizing idea: a ledger system specifies a preservation relation between record and derived value and checks it, while the compaction prompts describe intended fidelity to the record without supplying a procedure that checks whether the generated summary satisfies those requirements. The checks below are attempts to operationalize that missing relation, which makes them a consequence of the earlier distinction and not an independent list. Four checks are natural, and none depends on the details of a particular model.

The first is reference-bearing summaries. Each factual assertion in a handoff carries a pointer to the portion of the record offered in its support, so that a claim is never held without its provenance. This converts the summary from a free-standing text into an index into the record, and it makes omission of provenance a detectable defect and not an invisible one. Provenance models such as PROV-DM provide an established vocabulary for representing entities, activities, and derivations [16], and the database provenance literature supplies further distinctions for the relation to be checked [3]. A pointer makes a claim traceable but does not establish that the cited passage supports it. Evaluations of citation-generating language models distinguish answer correctness from citation quality [13], and handoff evaluation should likewise check whether the referenced evidence supports the assertion, including its scope and qualifications.

The second is a round-trip audit. Question-based consistency evaluation provides a method-

ological precedent: QAGS compares answers obtained from a summary and from its source to detect factual inconsistencies [19]. The proposed audit extends this approach to agent continuation with archive access and requires record-derived questions, so that evaluation can probe material absent from the summary and not only the claims it retains. The extension is not trivial. Continuation questions can concern actions and constraints and not only propositions, and archive access changes what the source is, since the agent can consult the record while answering. A set of questions about the earlier history is generated from the record, including questions about qualifications, exceptions, and failed attempts, and answered twice: once from the full history and once from the compacted state with archive access. The full-history answer is a comparison condition and not an unquestioned reference, since both configurations can make the same mistake. Where possible, each question should have an answer checked directly against identifiable record spans, and disagreements as well as shared errors should be evaluated against those spans. The questions must be generated from the record and not from the summary, since a question written from the summary inherits its omissions. Because the audit can be run at successive compaction depths, it is also the natural instrument for measuring the direction and amplification of drift. If answers about qualifications degrade while answers about conclusions persist, the audit detects selective loss of qualification information, and a consistent pattern across questions and across values of k, the number of compactions performed, distinguishes it from noise. To establish upward drift in apparent certainty, the audit must additionally show that later answers assert conclusions more categorically, or apply them beyond their recorded scope, and not that they express uncertainty or abstain.

The third is a constraint persistence test. A constraint introduced early in a task, such as a prohibition or an assumption the user stated, is tracked across repeated compactions to see whether it remains operative in later actions. Constraints are a good probe because their loss often produces no immediate failure; the agent simply acts as though the constraint had never existed. A public benchmark for a coding agent offers a related intervention, re-injecting constraints that carry provenance after context compaction, and reports a gain of 10.2 percentage points on SWE-bench Verified over 49 paired instances [11]. Re-injection maintains constraints and does not itself check that a summary's assertions are supported by their cited evidence, so it bears on this test and not on the first. The project also reports a repeat run across seeds on a 17-instance subset in which none of seven load-bearing instances recurred. That is an internal repeatability check by the project and not an independent replication, and the project is software documentation that has not been peer reviewed. It is best read as a precedent for the test and as an instance of why repeated runs matter.

The fourth is contrastive omission detection. The compacted state is compared with a variant from which a single material qualification has been removed, and the agent's behavior on questions that turn on that qualification is observed. Unchanged behavior does not by itself establish that the qualification was absent or unused, since the agent may recover it from the

archive or obtain equivalent information elsewhere. Comparing conditions with and without archive access helps distinguish reliance on the working state from successful recovery of omitted evidence, and thereby provides evidence about whether the qualification is represented, whether the agent uses it, and whether retrieval repairs its omission.

This essay reports no implementation of these checks and no new evaluation of the agents studied by Belcak et al., and it offers the checks as a specification of what would need to be measured. They share a common purpose: each evaluates fidelity to recorded evidence or constraints beyond success on the original scored objective, whether by checking the handoff itself or by testing the continuations it supports.

7 Limits of the Argument

Three limits should be recorded. First, the distinction drawn here does not show that the compaction results are unreliable; it shows that they are silent on a further question. A scored task may be one for which the working state is exactly sufficient, in which case preservation beyond the task's needs would be wasted. The criterion of warranted continuation matters where later questions are unforeseen, and many deployments do not have that character. Second, the proposed checks have costs of their own, since the round-trip audit requires running the full history, which is the expense compaction was meant to avoid. The audit would plausibly be run on samples and not continuously, and the sampling design is itself a research question. Third, the essay's analysis rests on the paper's reported results and prompts and not on independent replication, and the figures cited above are as the paper reports them.

8 Conclusion

The empirical result is that a model can continue a substantial task from a compact working state when the underlying material remains accessible. This supports compact working states backed by persistent resources, and it leaves the preservation criterion comparatively weak. The scored benchmarks measure whether enough survived to obtain the result. A stronger standard asks whether enough survived to support warranted continuation under changed questions and contested conclusions, and whether the evidence behind an earlier decision can still be recovered when someone demands it. The checks proposed above would also measure the direction and amplification of any drift in apparent certainty across successive compactions, which the scored benchmarks cannot reveal. These are different questions, and the answer to the first does not settle the second. The two projects are compatible: the architecture the paper demonstrates is a reasonable place to build verification, and verification is what the architecture, as demonstrated, does not yet provide.

References

[1] Yuri Kuratov, Aydar Bulatov, Petr Anokhin, Ivan Rodkin, Dmitry Sorokin, Artyom Sorokin, and Mikhail Burtsev. *BABILong: Testing the Limits of LLMs with Long Context Reasoning-in-a-Haystack*. Advances in Neural Information Processing Systems, Datasets and Benchmarks Track, 2024. arXiv:2406.10149.

[2] Yushi Bai et al. *LongBench v2: Towards Deeper Understanding and Reasoning on Realistic Long-Context Multitasks*. Proceedings of the Annual Meeting of the Association for Computational Linguistics, 2025.

[3] James Cheney, Laura Chiticariu, and Wang-Chiew Tan. *Provenance in Databases: Why, How, and Where*. Foundations and Trends in Databases, 1(4):379–474, 2009.

[4] Scott A. Crosby and Dan S. Wallach. *Efficient Data Structures for Tamper-Evident Logging*. Proceedings of the 18th USENIX Security Symposium, 2009.

[5] Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik R. Narasimhan. *SWE-bench: Can Language Models Resolve Real-World GitHub Issues?* International Conference on Learning Representations, 2024. arXiv:2310.06770.

[6] Greg Kamradt. *Needle in a Haystack: Pressure Testing LLMs*. Public code repository, 2023. No formal paper.

[7] Mike A. Merrill, Alexander G. Shaw, Nicholas Carlini, et al. *Terminal-Bench: Benchmarking Agents on Hard, Realistic Tasks in Command Line Interfaces*. International Conference on Learning Representations, 2026. arXiv:2601.11868.

[8] OpenAI. *Introducing SWE-bench Verified*. Published August 2024.

[9] OpenAI. *Why SWE-bench Verified No Longer Measures Frontier Coding Capabilities*. Published February 23, 2026.

[10] Bruce Schneier and John Kelsey. *Cryptographic Support for Secure Logs on Untrusted Machines*. Proceedings of the 7th USENIX Security Symposium, 1998.

[11] SaravananJaichandar (repository maintainer). *World Model MCP: Enforcement, Provenance, and Harness-Neutral Memory for AI Coding Agents*. Software documentation, versions 0.9.0–0.9.2, 2026; see benchmarks/repeat-mistake in the project repository.

[12] Martin Fowler. *Event Sourcing*. 12 December 2005. `https://www.martinfowler.com/eaaDev/EventSourcing.html`.

[13] Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. *Enabling Large Language Models to Generate Text with Citations*. Proceedings of EMNLP, pp. 6465–6488, 2023.

[14] Wojciech Kryscinski, Bryan McCann, Caiming Xiong, and Richard Socher. *Evaluating the Factual Consistency of Abstractive Text Summarization.* Proceedings of EMNLP, pp. 9332–9346, 2020.

[15] Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. *Lost in the Middle: How Language Models Use Long Contexts.* Transactions of the Association for Computational Linguistics, 12:157–173, 2024.

[16] Luc Moreau and Paolo Missier, editors. *PROV-DM: The PROV Data Model.* W3C Recommendation, 30 April 2013. `https://www.w3.org/TR/2013/REC-prov-dm-20130430/`.

[17] Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. *MemGPT: Towards LLMs as Operating Systems.* arXiv:2310.08560, version 2, 2024.

[18] Linda J. Skitka, Kathleen L. Mosier, and Mark Burdick. *Does Automation Bias Decision-Making?* International Journal of Human-Computer Studies, 1999.

[19] Alex Wang, Kyunghyun Cho, and Mike Lewis. *Asking and Answering Questions to Evaluate the Factual Consistency of Summaries.* Proceedings of ACL, pp. 5008–5020, 2020.

[20] Peter Belcak, Saurav Muralidharan, Jindong Jiang, Yonggan Fu, Wonmin Byeon, Hanrong Ye, Ryan Angilly, Ali Taghibakhshi, Shih-Yang Liu, Siru Ouyang, Greg Heinrich, and Pavlo Molchanov. *You Don't Need Long Context.* arXiv preprint arXiv:2611.05306, version 1, 2026.