International Journal of Scientific Research & Engineering Trends
Volume 12, Issue 4, July-Aug-2026, ISSN (Online): 2395-566X

The Completeness Gap: Coverage Failures at Every Layer of an Agentic LLM Research Pipeline

Ritik Sharma, Balvir Singh Thakur
Department of Computer Science & Engineering, University Institute of Technology (UIT), Himachal Pradesh University
Summer Hill, Shimla, Himachal Pradesh, India

Abstract — General-purpose large language models (LLMs) are increasingly deployed as autonomous research agents: given a web-search tool and a task-specific instruction, they are expected to compile structured, factually complete documents from public sources with little human oversight. We report an empirical case study, spanning three successive system architectures of a deployed research-automation module, evaluating whether a GPT-5-class model can reproduce the completeness and accuracy of a curated reference document. We show that a completeness gap recurs at every layer of the pipeline, not just at the LLM layer intuition would blame first. At the LLM layer, a self-directed search-stopping criterion does not scale with true task size, producing near-complete coverage on a low-activity subject but only 9 of 16 known events (56%) on a high-activity subject under comparable search effort. On a large structured disclosure table, the model correctly reports the verified aggregate figure but declines to enumerate individual line items it cannot verify from search snippets, capturing 0% of line items in the case measured. Replacing the model's self-directed search with a deterministic, non-LLM pre-fetch of an authoritative source, combined with a mandatory reconciliation checklist, more than triples measured coverage (9 to 28 events) — but building that deterministic stage itself surfaced two further completeness bugs, both caught and fixed before production through direct re-verification against live source data: a naive entity-name query missing 4 of 5 known related records until query construction was corrected, and a silent 1,000-record pagination cap that would have discarded 69% of one subject's true history had it gone undetected. We then report the most severe instance of the same underlying pattern: in a substantially more deterministic third-generation architecture, purpose-built to eliminate LLM-driven table extraction entirely, a "most recently filed wins" version-selection heuristic silently selected an incomplete partial-amendment filing over a complete original filing for the same reporting period, capturing only 1 of 10 real holdings (3.8% of true portfolio value) — a bug caught only via comparison against a second, independent reference for a different subject, not by the team's own prior testing, and which in turn silently disabled a downstream data-quality safety check that depended on the same broken data. We report these as an empirical case study rather than a statistically powered benchmark, and specify the exact follow-up measurement needed to convert the paper's central open finding into a confirmed before/after result.

Keywords— retrieval-augmented generation, agentic search, LLM evaluation, completeness, hallucination, information extraction, deterministic retrieval, pipeline reliability

I. INTRODUCTION

Large language models are now commonly deployed not just to answer isolated questions but to act as autonomous research agents — issuing their own web searches, deciding what evidence is sufficient, and synthesizing findings into a final structured document with minimal human review. This shift changes what "correctness" means for evaluation. Classic LLM evaluation asks: is what the model said true? Agentic research evaluation must additionally ask: did the model find everything it should have? The second question — completeness, or coverage — is harder to evaluate and, we argue, systematically under-tested relative to the first.

We study this gap through a task that is already in production use, not a synthetic benchmark: a module of a deployed analyst-facing research platform that automatically generates a structured profile document describing a Subject's history of documented Engagements with other entities, its current holdings, its key personnel, and its recurring strategic patterns (Section III defines this terminology precisely, in generic form). Because this module sits alongside several other analyst-facing tools in daily use, the completeness and accuracy gaps documented in this paper are not abstract evaluation-set artifacts; they are gaps a working analyst would actually encounter, which is part of why the underlying investigation insisted on exact, individually verified numbers rather than aggregate impressions of quality.

The intuitive fix for an unreliable, self-directed LLM search process is to remove the LLM from the parts of the task that most need to be exact, and replace it with deterministic, non-LLM retrieval against an authoritative source. This paper's central finding is that this intuition is necessary but not

sufficient. We document the same underlying failure — incomplete coverage of a known, enumerable ground truth — recurring at three separate layers of an evolving pipeline, each time for a structurally different reason, and each time discovered only because the investigating team deliberately re-verified system output against live source data rather than trusting logic that looked correct on inspection:

- At the LLM search layer, a self-directed stopping criterion does not scale with true task size (Section V-A).
- At the LLM extraction layer, the model correctly refuses to fabricate detail it cannot verify, which is honest model behavior but an incomplete system output (Section V-B).
- At the deterministic retrieval layer intended to fix (1) and (2), two further completeness bugs were found and fixed before production (Section V-C), and — in a more advanced, substantially more deterministic third-generation architecture purpose-built to eliminate the LLM entirely from table extraction — a third, more severe version-selection bug was found only after deployment, via comparison against a second independent reference (Section V-H).

Contributions. This paper contributes: (1) a documented, measured case of agentic search coverage failing to scale with true task size, distinguished from ordinary hallucination or precision failure; (2) a documented case of a "correct refusal, functional failure" pattern in structured data extraction; (3) an architectural pattern — deterministic ground-truth pre-fetch plus mandatory reconciliation checklist — with a measured before/after effect (9 to 28 events); (4) two further, pre-production-caught completeness bugs within that same deterministic layer, illustrating that a fix for an LLM-layer problem is not automatically correct by virtue of containing no LLM; (5) the paper's most severe and most counter-intuitive finding: a substantially more deterministic third architecture generation, purpose-built to close the LLM's table-extraction gap, still failed on real data due to an unsafe "recency implies completeness" assumption in document-version selection, and this failure silently disabled an unrelated downstream safety mechanism; (6) a report of the exact follow-up experiment needed to convert this paper's central open finding from a confirmed "before" into a measured "before/after" result; and (7) an honest accounting of this work's evidentiary strength as a small-scale case study across three system generations, rather than a controlled benchmark.

II. RELATED WORK

1. Faithfulness and completeness of web-augmented LLMs
Tool-use architectures that let an LLM issue its own search queries and cite retrieved evidence trace to WebGPT [1], which fine-tuned GPT-3 to browse and answer long-form questions with citations, evaluated primarily on factual accuracy of what was said. Subsequent long-form factuality work has largely continued this emphasis on precision of claims: SAFE, evaluated via the LongFact benchmark [3], uses live search itself as an automated fact-checker, decomposing a response into atomic claims and verifying each independently. Huang et al.'s survey of LLM hallucination [4] formalizes a taxonomy separating factuality hallucination (contradicting world knowledge) from faithfulness hallucination (deviating from provided context or instructions) — a useful vocabulary here, because the tabular-extraction failure we report (Section V-B) fits neither category cleanly: the model's aggregate figure is factually correct and its refusal to enumerate rows is faithful to its own uncertainty, yet the system's output is still incomplete. Most directly relevant is BrowseComp [2], a 2025 OpenAI benchmark of questions requiring persistent navigation to locate "hard-to-find, entangled" facts, which shows browsing-capable models performing poorly specifically when information is scattered and effortful to assemble, rather than simply absent from the model's training data. Our high-activity-subject failure (Section V-A) is best read as a real-system instance of the same underlying difficulty, measured as coverage of a known finite set rather than accuracy on single-answer questions.

2. Coverage and recall in retrieval-augmented generation
A separate line of RAG evaluation work has begun explicitly decoupling retrieval completeness from generation faithfulness, rather than measuring end-to-end QA accuracy alone. RGB [5] tests LLMs on "information integration" — whether a model correctly assembles an answer requiring combination of multiple, separately retrieved facts — and shows this capability degrading well before simple single-fact retrieval does. RAGChecker [6] introduces "claim recall" as a retrieval-side diagnostic metric distinct from generation-side hallucination metrics, formalizing the distinction this paper relies on: a system can fail by retrieving too little (recall) independently of whether it fabricates anything (precision). CRUD-RAG [7] similarly argues task-completeness categories, not answer plausibility, should anchor RAG evaluation. RECALL [8] shows LLMs are frequently misled by incomplete or conflicting retrieved context rather than correctly flagging the gap — relevant to our Section V-B finding that a correct refusal-to-guess is itself a distinct, better-behaved failure mode than the silent incompleteness this line of work usually measures. None of this literature, to our knowledge, measures whether retrieval coverage scales correctly with the true size of the underlying answer set — the RAG-completeness metrics above are computed per-question or per-document, not as a function of how much there objectively is

to find for a given subject. This is the specific gap our Section V-A finding addresses.

3. Agentic search behavior and stopping criteria

A substantial body of work addresses when an LLM agent should retrieve, within the ReAct [9] paradigm of interleaved reasoning and tool calls. Self-RAG [10] trains a model to emit reflection tokens that decide, on demand, whether further retrieval is needed. FLARE [11] triggers retrieval proactively based on generation-time uncertainty. IRCoT [12] interleaves retrieval with chain-of-thought reasoning for multi-hop questions, improving recall over single-shot retrieval. More recently, Search-R1 [13] uses reinforcement learning to train a model to interleave reasoning and search-engine calls, including implicitly learning when to stop. All of these systems make a stopping decision, but evaluate it via downstream QA accuracy on fixed benchmarks — none, to our knowledge, directly studies whether the number of retrieval steps an agent performs scales with the true cardinality of the evidence needed, which is the specific behavior we measure in Section V-A.

The behavioral pattern we observe — search effort that feels adequate to the agent but does not scale with true task size — has a name in decision theory: satisficing, Herbert Simon's classic account of bounded-rational agents stopping at "good enough" rather than optimal solutions. Chehade et al. [14] recently formalized a satisficing, threshold-based (rather than maximizing) stopping rule for LLM inference-time behavior, giving a peer-reviewed LLM-specific anchor for this framing. The underlying phenomenon is older than LLMs: Prabha et al. [15] show empirically that human searchers decide they have "enough" information well before exhausting available sources, and Maxwell et al. [16], in a large-scale search-stopping-rules study, show human stopping decisions correlate only weakly with the true richness of the topic being searched. Our contribution is to show the same qualitative pattern reproducing in an LLM agent operating over a task with an externally verifiable ground-truth cardinality, which the cited human-behavior studies could only estimate indirectly.

4. Structured and tabular data extraction in financial and web-scale sources

Because Finding 2 and Finding 3 both concern exhaustive, correct transcription of a large structured table, we situate them against the established financial-table-extraction benchmark lineage: FinQA [17] and its conversational extension ConvFinQA [19] test multi-step numerical reasoning over report tables, with reported model accuracy well below human-expert baselines, collapsing further as the number of required reasoning steps increases; TAT-QA [18] specifically requires cell-level extraction and arithmetic over hybrid table-and-text financial content. Most directly relevant to Finding 2 is FinanceBench [20], which reports that a strong retrieval-augmented model "incorrectly answered or refused to answer" a large majority of questions posed directly against real regulatory filings — consistent with our observation that correct aggregate-level reporting does not imply correct or complete line-item-level extraction.

Two 2025 benchmarks probe this gap more directly: FinTagging [21] finds LLMs handle coarse entity/number extraction from filings well but perform substantially worse at fine-grained, per-line-item structuring against a formal taxonomy, and SECQUE [22] evaluates comparison and ratio-calculation tasks directly against filing text. Outside the financial domain, OTT-QA [23] requires open-domain fusion of table and free-text evidence rather than extraction from one clean source document, closely paralleling our scenario of assembling a table from scattered search-result snippets; TARGET [24] isolates table-retrieval quality specifically and shows it degrades sharply once structural metadata (e.g., table titles) is unavailable — a plausible mechanism for why our model, working only from search snippets, judged individual rows unverifiable while trusting the aggregate figure reported in prose.

5. Document versioning and authoritative-source selection

Finding 3 (Section V-H) concerns a deterministic retrieval bug: selecting an incomplete amendment filing over a complete original filing under a naive recency heuristic. We found substantially less directly relevant prior work here. The nearest analog identified, WithdrarXiv [25], builds a large dataset of retracted preprints and classifies why they were withdrawn, but does not address automated systems mis- selecting an incomplete document version during retrieval. We were unable to locate a benchmark or study evaluating automated correctness of original-versus-amendment selection in structured public filing systems specifically. We report this as a gap our third finding speaks to, noting a negative literature search is suggestive rather than dispositive of true novelty.

III. SYSTEM AND TASK

Terminology (anonymized). To avoid identifying the underlying commercial system, we describe the task using generic terms, used consistently throughout the rest of the paper:

Generic term	Meaning
Subject	The entity being profiled
Engagement	A single discrete historical interaction between the Subject and another entity
Filing Index	The authoritative, structured public record system in which Engagements and holdings are documented by regulation or convention
Profile	The final structured output document: Engagement history, current holdings, associated personnel, and recurring patterns

Task. Given a named Subject, produce a Profile summarizing: (a) its historical Engagements, (b) its current holdings as disclosed in the Filing Index, (c) key personnel associated with the Subject, distinguishing internal leadership from externally nominated individuals, and (d) recurring strategic patterns across Engagements. This is not a synthetic evaluation task: it is one module of an already-deployed platform used by analysts for research, engagement prioritization, and related structured-document tasks, alongside several other analyst-facing tools already in production.

Why this task is a useful stress test. Three properties make failure modes measurable rather than anecdotal: (1) ground truth for Engagement count is enumerable, because Filing Index entries are structured and independently queryable; (2) the task rewards recall and precision independently, so a Profile can be complete-but-sparse or rich-but-partially-fabricated, and these are distinguishable; (3) independently produced reference Profiles exist for direct comparison.

Evaluation methodology. Every completeness or accuracy claim in this paper is backed by one of two kinds of evidence: (i) comparison against an independently produced reference Profile for the same real-world Subject, generated via the model family's normal interactive interface and treated as an achievable quality bar; or (ii) direct, live re-verification against the Filing Index itself — re-fetching and re-counting the real underlying records, rather than trusting either system's self-report. Several findings in this paper (notably Sections V-C and V-H) were reached only because output was checked against live source data a second time, after an initial inspection suggested the responsible component "looked correct." This distinction — inspecting logic that appears correct versus reproducing the actual failure against real data— recurs throughout this paper: a component can be architecturally sound and still be silently wrong on real inputs.

1. Generation 1 — naive single-call API replication

The baseline architecture is a single API call: one long, detailed system prompt — structurally equivalent to a "custom assistant" configuration a user would set up in a consumer chat interface — instructing the model to research a named Subject "from scratch" using an attached live web-search tool, and to emit findings as one structured JSON document against a fixed schema. No other data source is provided; the model's own search judgment is the sole retrieval mechanism, and its own judgment governs when to stop searching. This is deliberately the simplest architecture a practitioner would try first, and is the baseline the rest of this paper measures improvement against.

IV. FINDINGS: THE LLM SEARCH AND EXTRACTION LAYERS

1. Finding 1 — Search effort does not scale with true task size

On a low-activity Subject (a small historical footprint), Generation 1's self-directed search achieved coverage close to an independently verified reference Profile's own Engagement list for that Subject. On a high-activity Subject (a substantially larger historical footprint), the same architecture, using a comparable volume of search activity, captured only 9 of 16 known Engagements (56%) against an independently verified reference list.

The model was not failing to search — it issued a plausible number of search queries and then reached its own subjective judgment that it had done enough, and that judgment did not scale with how much there actually was to find. We characterize this as a satisficing failure (Section II-C): without an external signal of true task size, a self-directed stopping criterion under-covers precisely the Subjects where coverage matters most — the busiest, most complex ones — while appearing to perform well on simpler cases. A system evaluated only on the easy, low-footprint case would look reliable; evaluated only on the hard, high-footprint case, it would look badly broken. Neither evaluation alone reveals the actual failure mode: coverage degrades as true task complexity increases, which is the inverse of what is wanted from a system meant to save analyst time on exactly the busiest, most complex Subjects.

This generalizes beyond this task: any agentic system whose search-stopping decision is made entirely by the agent itself, with no external signal for how much ground truth exists, is structurally prone to this pattern.

2. Finding 2 — Correct refusal, functional failure, in structured table extraction

One recurring sub-task required transcribing a Subject's complete holdings disclosure — a structured table with

potentially dozens of individual line items (security name, ticker, dollar value, share count) — from a Filing Index disclosure. Generation 1 reliably found and correctly reported the table's aggregate total value. It did not reliably enumerate the individual line items.

In the case measured, this produced 0% coverage of individual holdings line items ($0 of a multi-billion-dollar aggregate value broken into rows) despite a correctly reported aggregate total; the model explicitly stated, in its own output, that it could not confidently enumerate the underlying rows from search-result snippets, and declined to guess rather than fabricate them. A comparison case, for a Subject with a much smaller and simpler holdings table, was captured completely and correctly by the same mechanism.

This is not hallucination — the model behaved exactly as instructed (never fabricate; state uncertainty rather than guess) — but it is a functional failure of the system as a whole: a user receiving this Profile gets a correct headline number and an empty table. We interpret this as a capability boundary that scales with the target dataset's real-world size and complexity (Section II-D), not a matter of model willingness or instruction-following, and note it requires a structurally different fix than Finding 1: better prompting cannot manufacture verifiable per-row evidence the model does not have. A later root-cause investigation (Section V-H) found a second, structurally unrelated cause of the same surface symptom — "holdings table incomplete" — in a different, more deterministic architecture. This recurrence is itself a methodological point worth preserving: symptom-level pattern-matching across systems can mislead if root causes are not independently re-verified each time.

V. FINDINGS: THE DETERMINISTIC RETRIEVAL LAYER

1. Intervention 1 — Deterministic pre-fetch and mandatory reconciliation checklist

To address Finding 1, a separate, fully deterministic process — no LLM involved — was introduced to query the Filing Index directly before the LLM call, enumerating every entity the Subject has a documented public relationship with, using only structured filing metadata (form types, dates, involved-party names) independent of any LLM judgment. This ground-truth list is injected into the LLM's prompt as a mandatory checklist: for every listed item, the model must either produce a corresponding Profile entry or explicitly state its reason for excluding it.

On the high-activity Subject from Section IV-A, this raised measured coverage from 9 to 28 well-evidenced Engagements — more than triple the unaided baseline — while every included or excluded item remained individually justified in the output rather than silently dropped. The general pattern: move completeness responsibility from the model's self-directed judgment to a deterministic retrieval step, and reserve the LLM for synthesis, narrative construction, and surfacing Engagements that have no Filing Index paper trail at all (Section V-C), which deterministic retrieval structurally cannot see.

2. Two completeness bugs caught before production, within the deterministic layer itself

Building this deterministic pre-fetch stage surfaced two further completeness bugs — evidence that a fix built specifically to correct an LLM-layer completeness problem is not automatically correct simply because it contains no LLM. Both were caught by deliberately reproducing the retrieval step against live data and manually checking the result count against independently known ground truth, before the mechanism was trusted in production — not by code review alone.

Query construction. The checklist mechanism's core assumption — that a single query against the Filing Index would find every related entity — was itself found to be false in its first implementation. A naive query using a Subject's full formal name found only 1 of 5 known related filing entities for a real test Subject; a corrected query strategy, based on extracting the Subject's single most distinctive identifying term rather than its full name, found 5 of 5.

Silent pagination truncation. A related infrastructure-level bug surfaced in the same verification pass: the Filing Index's own API silently truncates results after a fixed count— 1,000 records, in the case tested — unless a pagination mechanism is explicitly used to continue past that limit. For one heavily used real-world test entity, the untruncated true record count was 3,255; a naive, non-paginating query would therefore have silently returned only 31% of that entity's true history, with no error and a plausible-looking result set. This is a mundane but consequential integration detail: a deterministic retrieval step is only as complete as its awareness of its own source system's pagination behavior, and this class of failure looks like success — no error, a plausible result count — while being badly incomplete.

3. Intervention 2 — Schema fields as a free extraction lever

Independently of Intervention 1, adding explicit output-schema fields for phenomena the model was already implicitly discovering during search, but had no structured place to

record, measurably improved captured information with no new data source or search capability. Fields added: one distinguishing a Subject's internal leadership from externally nominated individuals it had placed on other companies' boards; one for an Engagement's concrete final outcome, kept separate from its stated opening objective; and one explicitly capturing situations where a different entity held the primary public evidentiary trail for an Engagement and the Subject was a secondary or supporting participant (a "second-mover" or coalition situation).

The underlying information was already being found during the model's research; it was being discarded at the synthesis step for lack of a place to put it. We report this separately from Intervention 1 because it is a materially cheaper class of fix — it requires no new architecture or retrieval infrastructure, only schema design informed by direct comparison against a richer independent reference.

4. A full end-to-end measured run

With the checklist mechanism, the expanded schema, and general prompt refinements combined, one full live end-to-end run was measured against the prior unaided baseline for the same high-activity Subject used throughout Section IV-A and Section V-A:

Table 1. Generation 1/2 Baseline vs. Combined Interventions 1+2 (same Subject)

Metric	Baseline (unaided)	With Interventions 1+2
Engagement entries produced	9	28
Cross-Engagement analytical observations	field not present in schema	3 populated entries
Source documents cited	not separately measured	35
Personnel / nominee records	not separately measured	8

A known data-quality issue is also worth reporting for completeness: two entries in the resulting Engagement list referred to the same real-world event in a way that suggested near-duplication, and were flagged for manual cleanup rather than silently accepted.

5. What deterministic retrieval structurally cannot see

Intervention 1's checklist mechanism is fundamentally limited to entities with a documented filing relationship to the Subject in the Filing Index. Some legitimate Engagements have no such filing at all — for example, an entity that publicly voices support for a different activist's campaign, or quietly builds a position later reported by journalists to be a source of private pressure, without ever filing the disclosure that would place it on a structured checklist.

The system was explicitly instructed to treat the checklist as a floor, not a ceiling: required to cover everything on the deterministic list, but still expected to use its own web-search judgment to find Engagements outside that list — particularly these no-paper-trail coalition or support situations — and to mark them with a distinct, lower-confidence label rather than either omitting them or asserting them with the same confidence as a filing-backed event. In the one full end-to-end run measured (Section V-D), this distinct code path was not observed to trigger — every Engagement entry produced came back at the higher-confidence, filing-backed tier, none at the coalition/no-filing tier. We report this as an honest non-result rather than omitting it: the mechanism exists and was exercised by design, but its real-world effectiveness is, on the evidence available, unverified rather than confirmed working.

VI. FINDING 3: A MORE DETERMINISTIC ARCHITECTURE, AND A NEW, MORE SEVERE BUG

1. Generation 3 — a substantially more deterministic architecture

A separate, more architecturally sophisticated system was also investigated. Rather than relying on the LLM's web-search tool for holdings data (Finding 2's root cause in Generation 1), this system fetched and parsed the Subject's holdings disclosure directly from the Filing Index's own structured data file — a fully deterministic step with no LLM involvement, in principle immune to Finding 2 entirely. It also used the Filing Index's full-text search to deterministically discover every entity the Subject has a documented filing relationship with, built a per-target filing chronology (splitting a Subject's repeated Engagements with the same entity, years apart, into separate episodes using a gap-length heuristic), and layered multiple staged web-research passes on top for anything a purely deterministic approach could not see — including a dedicated second pass whose only job was to hunt for Engagements missing from the first pass's output, and a dedicated personnel-research pass. It also included a deterministic post-processing safety check: comparing the Subject's largest current holdings against the discovered Engagement list, and flagging any large, uncovered holding as worth manual review, on the theory that a large position with no identified public Engagement might indicate an under-the-radar situation the research passes had missed.

This is architecturally the most advanced of the systems investigated, and on paper should have been immune to Finding 2 — the deterministic structured-holdings fetch was specifically designed to close that exact gap. It was not.

2. The bug: deterministic retrieval selecting the wrong version of an authoritative document

When compared against an independently produced reference Profile for a different real-world Subject than any used earlier in this investigation, this system's holdings data was found to be severely incomplete: it captured only 1 of 10 real holdings, representing 3.8% of the Subject's true reported portfolio value, versus the reference Profile's complete and independently verified 10-of-10 capture.

This was root-caused, not merely observed, by directly re-fetching the underlying Filing Index records live: the Subject had filed two versions of its holdings disclosure for the same reporting period — an original filing containing the complete 10-item table, and a later amendment to the same period containing exactly 1 item, evidently a partial correction adding one late-disclosed position rather than a full restatement. The system's selection logic picked the amendment purely because it was the more recently filed of the two, without checking whether it was a complete restatement or only a partial correction. Direct, live re-fetching and counting of both documents' contents confirmed the row counts precisely (10 vs. 1) before this was accepted as the confirmed root cause.

This finding is direct evidence against an intuitive but false assumption: that removing the LLM from a pipeline stage removes its risk of being wrong. The deterministic stage here had zero hallucination risk — it was mechanically parsing a real document — and was still severely wrong on real data, because "most recently filed" is an unsafe proxy for "most complete" whenever multiple authoritative versions of the same real-world record can exist. This is exactly the kind of distinction an implementer might not think to check for, precisely because it "feels like" a problem only an LLM-based stage could have.

3. Failure propagation to a downstream safety mechanism

This system's own under-covered-holdings safety check (Section VI-A) was itself downstream of the broken holdings data: with only 1 of 10 real positions visible to it, it had no opportunity to flag the other 9 for review — including a position that independently verified reference data showed had a real, live Engagement attached to it with no regulatory filing at all (a coalition-style situation, Section V-E). This is precisely the kind of situation the safety mechanism existed to catch, and it could not, because its input was silently compromised upstream. This is a clean illustration of failure propagation: one upstream data-completeness bug silently disabled a downstream safety mechanism that had no way of knowing its own input was compromised.

4. Secondary findings from the same comparison

Two smaller, independently confirmed findings from the same comparison are useful secondary data points, though less central to this paper's main argument. First, a literal duplicate record — the same event emitted twice, identically — survived to final output despite a deduplication step existing earlier in the pipeline for an intermediate stage; the final synthesis stage had no deduplication pass of its own. Second, the same category-segregation and outcome/coalition-analysis schema gaps described for Generation 1/2 (Section V-C) were independently found to also be present in this architecturally distinct Generation 3 system, suggesting these particular schema gaps are a property of how the task was specified across systems, not an artifact of one particular implementation.

5. Status of this finding

This finding is root-caused and verified against live source data, as described above, but the fix has not yet been implemented or re-measured. Unlike Section V-A's Intervention 1 result, which has a real, measured before/after (9 to 28 Engagement entries), this section currently has a confirmed "before" but no "after." We treat this explicitly as the paper's central open item rather than rounding it up to a claimed fix; Section VIII specifies the exact follow-up experiment needed to close it.

VII. DISCUSSION

Read together, these findings suggest a more specific design principle than "use RAG" or "add tools": completeness and correctness require different kinds of guarantees, and neither is free just because the other has been addressed. Finding 1 shows that giving a capable model a search tool and a good prompt is not sufficient for completeness, because the model has no way to know how much there is to find — a structural property of self-directed stopping criteria (Section II-C), not a prompting deficiency. Finding 2 shows that even a well-behaved model that correctly refuses to fabricate can still produce an incomplete system output, meaning "the model didn't hallucinate" is an insufficient bar for system correctness in extraction-heavy tasks.

The paper's central, recurring theme is what happens next. Intervention 1 fixes Finding 1 by removing the LLM from the completeness-critical decision entirely — and building that fix surfaced two further completeness bugs (Section V-B), both caught before production by deliberately re-verifying output

against live data rather than trusting logic that read correctly on inspection. Generation 3 goes further, removing the LLM from holdings extraction entirely — and still failed, more severely than either LLM-layer finding, because "most recently filed" is an unsafe proxy for "most complete" (Section VI-B). Across three separate instances — query construction, pagination, and document-version selection — the same lesson recurs at increasing severity: determinism removes hallucination risk, but it does not remove correctness risk. A deterministic pipeline stage needs its own explicit verification and testing discipline; correctness cannot be assumed simply because no LLM was involved in producing the output.

There is a second-order observation worth making explicitly, because it connects this paper's central finding to its own methodological limitations (Section IX). The two Generation-1/2 deterministic bugs (Section V-B) were caught before production, because the team deliberately re-verified the mechanism against live data for the specific test Subject at hand. The Generation-3 bug (Section VI-B) was not caught by the same team's own prior testing of that system — it surfaced only once output was compared against an independent reference for a different Subject, one that happened to have two versions of the same filing on record. In other words, testing against one Subject's data does not guarantee correctness against another Subject's data whenever the failure mode depends on a data pattern (here, duplicate filings for one period) that is present in some real-world records and absent in others. This is the same small-sample blind spot this paper's own limitations (Section IX) caution against, now shown operating inside the investigation itself, not just in the paper's evidentiary base.

Finally, the failure-propagation result (Section VI-C) generalizes beyond this task: a downstream safety or validation mechanism is only as reliable as its upstream inputs, and a mechanism that assumes its input is trustworthy has no way to detect that the assumption has been violated. This argues for testing multi-stage pipelines for cross-stage failure propagation specifically, not only for the correctness of each stage in isolation. Similarly, the recurrence of the same schema gaps across two architecturally unrelated systems (Section VI-D) suggests that some completeness gaps are properties of an underspecified task definition — what fields a Profile should contain — rather than of any one system's implementation, which argues for treating schema design as a first-class part of task specification rather than an implementation detail decided late.

VIII. THREATS TO VALIDITY AND LIMITATIONS

This is an empirical case study over a small number of real-world Subjects across three successive system architectures, not a large, randomly sampled, statistically powered benchmark. We report precise, individually verifiable per-case measurements rather than aggregate statistics with confidence intervals, and we do not claim the specific figures reported (e.g., 56% coverage, a 9-to-28 improvement, 3.8% portfolio-value capture) generalize quantitatively beyond the cases measured; a claim such as "this technique triples coverage in general" would require testing across many more Subjects and reporting aggregate statistics with confidence intervals.

Several findings carry their own, more specific caveats. The coalition/no-paper-trail detection mechanism (Section V-E) is architecturally present and intentionally designed, but was not observed to trigger in the one full end-to-end run measured; its real-world effectiveness is unverified, not confirmed working, and is described that way rather than claimed as a success. Section VI's central finding currently has a confirmed "before" measurement but no "after": the fix (preferring the more complete of two same-period filings rather than simply the more recent one) has not yet been implemented or re-measured, and we treat this as an open item rather than rounding it up to a resolved result. Cost and latency were observed to increase materially with each completeness-improving intervention — more search iterations, an added multi-minute deterministic pre-fetch stage — but were not rigorously quantified as a formal completeness-per-dollar or completeness-per-minute curve in this investigation.

All measurements were taken by the same team that built the systems under study, which is itself a limitation directly connected to this paper's own findings: as discussed in Section VII, that team's own prior testing did not catch the Generation-3 version-selection bug, and we cannot rule out other undetected issues of the same kind in either the baseline or the interventions reported here.

Future Work

The single most valuable follow-up measurement is closing Section VI's open item. Once the fix described there is implemented — preferring the more complete of two same-period authoritative filings, not simply the more recently filed one — the required experiment is a direct re-run on the same real-world Subject used in Section VI, measuring, in the same format as Table I:

Table 2. Proposed Generation 3 Before/After Measurement (template — post-fix values pending)

Metric	Before fix (measured)	After fix (pending)
Holdings captured (count)	1 of 10	*[to be measured; expected to move toward 10 of 10]*
Holdings captured (% of true portfolio value)	3.8%	*[to be measured; expected to move toward 100%]*
Downstream safety check flags the previously invisible large position with an independently confirmed, undocumented Engagement	Not applicable — position was invisible to the check	*[to be measured as its own line item, not assumed]*
Engagement-entry count	9 (unaided) / 28 (with Interventions 1+2)	*[to be measured; may not change — the missing Engagement here has no regulatory filing, so fixing the holdings bug surfaces it for manual/LLM review but does not automatically convert it into a confirmed Engagement entry without a further research pass]*

We deliberately report the third row as a distinct, causally connected result rather than folding it into the holdings-capture number alone, and the fourth row is phrased to avoid assuming the intuitive outcome — both should be reported as whatever is actually measured, not as an expected result confirmed in advance.

Beyond this, the natural extension of Sections IV-A and V-A is repeating the before/after comparison across a much larger, randomly sampled set of Subjects spanning a wider range of activity levels, and reporting aggregate coverage statistics with confidence intervals rather than per-case counts. A second useful extension, motivated directly by Section VII's discussion, is systematically auditing other deterministic pipeline stages for the same class of failure identified in Section VI — treating "which heuristic selects among multiple versions of the same record" as its own explicit verification target across many Subjects, rather than trusting a single successful test. Third, testing whether Intervention 2's schema-design lever (Section V-C) generalizes — whether systematically comparing model output against a richer independent reference and mining the difference for missing schema fields is a repeatable technique, or was specific to this task — would clarify how broadly applicable that low-cost intervention class is. Finally, formally quantifying the cost/latency-versus-completeness trade-off across all interventions reported here would let a practitioner choose an operating point rather than treating "more complete" as an unqualified improvement.

X. CONCLUSION

We presented an empirical case study, spanning three successive system architectures, of a GPT-5-class LLM-based research pipeline attempting to reproduce the completeness and accuracy of a curated structured-profile document. We found that a naive but reasonable LLM-only baseline fails in two independent, measurable ways — a self-directed search-stopping criterion that does not scale with true task size, and a structurally correct but functionally incomplete refusal to enumerate large structured tables from search snippets. We found that the natural fix — removing the LLM from the completeness-critical steps and replacing it with deterministic retrieval — is necessary but not automatically sufficient: building that fix surfaced two further completeness bugs caught before production, and a substantially more deterministic third-generation architecture, purpose-built to eliminate the LLM's role in table extraction entirely, still failed more severely than either LLM-layer finding, due to an unsafe "recency implies completeness" assumption that also silently disabled an unrelated downstream safety mechanism. The consistent lesson across all three layers is that determinism removes hallucination risk but not correctness risk, and that verification against live, real-world data — repeated across genuinely different Subjects, not just the one at hand — is what actually catches these failures, not code review or architectural sophistication alone. We report these as precise, individually verifiable findings from a small case study across three system generations, and specify the concrete next measurement needed to convert this paper's central open finding into a confirmed before/after result.

REFERENCES

1. R. Nakano, J. Hilton, S. Balaji, J. Wu, L. Ouyang, C. Kim, C. Hesse, S. Jain, V. Kosaraju, W. Saunders, X. Jiang, K. Cobbe, T. Eloundou, G. Krueger, K. Button, M. Knight, B. Chess, and J. Schulman, "WebGPT: Browser-assisted

question-answering with human feedback," arXiv:2112.09332, 2021.
2. J. Wei, Z. Sun, S. Papay, S. McKinney, J. Han, I. Fulford, H. W. Chung, A. Tachard Passos, W. Fedus, and A. Glaese, "BrowseComp: A Simple Yet Challenging Benchmark for Browsing Agents," arXiv:2504.12516, 2025.
3. J. Wei, C. Yang, X. Song, Y. Lu, N. Hu, J. Huang, D. Tran, D. Peng, R. Liu, D. Huang, C. Du, and Q. V. Le, "Long-form factuality in large language models," in Proc. NeurIPS, 2024. arXiv:2403.18802.
4. L. Huang, W. Yu, W. Ma, W. Zhong, Z. Feng, H. Wang, Q. Chen, W. Peng, X. Feng, B. Qin, and T. Liu, "A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions," ACM Trans. Information Systems, vol. 43, no. 2, Jan. 2025. arXiv:2311.05232.
5. J. Chen, H. Lin, X. Han, and L. Sun, "Benchmarking Large Language Models in Retrieval-Augmented Generation," in Proc. AAAI Conf. Artificial Intelligence, vol. 38, pp. 17754-17762, 2024. arXiv:2309.01431.
6. D. Ru, L. Qiu, X. Hu, T. Zhang, P. Shi, S. Chang, C. Jiayang, C. Wang, S. Sun, H. Li, Z. Zhang, B. Wang, J. Jiang, T. He, Z. Wang, P. Liu, Y. Zhang, and Z. Zhang, "RAGChecker: A Fine-grained Framework for Diagnosing Retrieval-Augmented Generation," in Proc. NeurIPS Datasets and Benchmarks Track, 2024. arXiv:2408.08067.
7. Y. Lyu, Z. Li, S. Niu, F. Xiong, B. Tang, W. Wang, H. Wu, H. Liu, T. Xu, and E. Chen, "CRUD-RAG: A Comprehensive Chinese Benchmark for Retrieval-Augmented Generation of Large Language Models," ACM Trans. Information Systems, 2024, doi:10.1145/3701228. arXiv:2401.17043.
8. Y. Liu, Y. Huang, S. Feng, X. Liu, and B. Qin, "RECALL: A Benchmark for LLMs Robustness against External Counterfactual Knowledge," arXiv:2311.08147, 2023.
9. S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, and Y. Cao, "ReAct: Synergizing Reasoning and Acting in Language Models," in Proc. ICLR, 2023. arXiv:2210.03629.
10. A. Asai, Z. Wu, Y. Wang, A. Sil, and H. Hajishirzi, "Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection," in Proc. ICLR, 2024. arXiv:2310.11511.
11. Z. Jiang, F. F. Xu, L. Gao, Z. Sun, Q. Liu, J. Dwivedi-Yu, Y. Yang, J. Callan, and G. Neubig, "Active Retrieval Augmented Generation," in Proc. EMNLP, 2023, pp. 7969-7992. arXiv:2305.06983.
12. H. Trivedi, N. Balasubramanian, T. Khot, and A. Sabharwal, "Interleaving Retrieval with Chain-of-Thought Reasoning for Knowledge-Intensive Multi-Step Questions," in Proc. ACL, 2023, pp. 10014-10037. arXiv:2212.10509.
13. B. Jin et al., "Search-R1: Training LLMs to Reason and Leverage Search Engines with Reinforcement Learning," arXiv:2503.09516, 2025.
14. M. Chehade, S. S. Ghosal, S. Chakraborty, A. Reddy, D. Manocha, H. Zhu, and A. S. Bedi, "Bounded Rationality for LLMs: Satisficing Alignment at Inference-Time," in Proc. ICML, 2025. arXiv:2505.23729.
15. C. Prabha, L. S. Connaway, L. Olszewski, and L. R. Jenkins, "What is enough? Satisficing information needs," Journal of Documentation, vol. 63, no. 1, pp. 74-89, 2007. doi:10.1108/00220410710723894.
16. D. Maxwell, L. Azzopardi, K. Jarvelin, and H. Keskustalo, "Searching and Stopping: An Analysis of Stopping Rules and Strategies," in Proc. ACM Int. Conf. Information and Knowledge Management (CIKM), 2015, pp. 313-322. doi:10.1145/2806416.2806476.
17. Z. Chen, W. Chen, C. Smiley, S. Shah, I. Borova, D. Langdon, R. Moussa, M. Beane, T.-H. Huang, B. Routledge, and W. Y. Wang, "FinQA: A Dataset of Numerical Reasoning over Financial Data," in Proc. EMNLP, 2021. arXiv:2109.00122.
18. F. Zhu, W. Lei, Y. Huang, C. Wang, S. Zhang, J. Lv, F. Feng, and T.-S. Chua, "TAT-QA: A Question Answering Benchmark on a Hybrid of Tabular and Textual Content in Finance," in Proc. ACL-IJCNLP, 2021. arXiv:2105.07624.
19. Z. Chen, S. Li, C. Smiley, Z. Ma, S. Shah, and W. Y. Wang, "ConvFinQA: Exploring the Chain of Numerical Reasoning in Conversational Finance Question Answering," in Proc. EMNLP, 2022, ACL Anthology 2022.emnlp-main.421.
20. P. Islam, A. Kannappan, D. Kiela, R. Qian, N. Scherrer, and B. Vidgen, "FinanceBench: A New Benchmark for Financial Question Answering," arXiv:2311.11944, 2023.
21. Y. Wang, L. Qian, X. Peng, Y. Ren, K. Wang, et al., "FinTagging: Benchmarking LLMs for Extracting and Structuring Financial Information," arXiv:2505.20650, 2025.
22. N. Ben Yoash, M. Brief, O. Ovadia, G. Shenderovitz, M. Mishaeli, R. Lemberg, and E. Sheetrit, "SECQUE: A Benchmark for Evaluating Real-World Financial Analysis Capabilities," in Proc. 4th Workshop on Generation, Evaluation and Metrics (GEM), ACL, 2025. arXiv:2504.04596.
23. W. Chen, M.-W. Chang, E. Schlinger, W. Y. Wang, and W. W. Cohen, "Open Question Answering over Tables and Text," in Proc. ICLR, 2021. arXiv:2010.10439.
24. X. Ji, P. Glenn, A. G. Parameswaran, and M. Hulsebos, "TARGET: Benchmarking Table Retrieval for Generative Tasks," arXiv:2505.11545, 2025.

25. D. Rao, J. Young, T. Dietterich, and C. Callison-Burch, "WithdrarXiv: A Large-Scale Dataset for Retraction Study," arXiv:2412.03775, 2024.