Context Integrity

A Benchmark for Long-Running AI Agent Memory and Action

Olajide Al-ameen Bad Theory Labs, Lagos Version 0.1 · July 2026

Abstract

AI agents are increasingly expected to operate across long-running workflows: reading documents, remembering user preferences, updating stale facts, retrieving evidence, and choosing actions. Existing evaluations usually isolate one piece of this system. Long-context benchmarks test whether a model can attend over a fixed prompt. Retrieval benchmarks test whether a relevant passage can be found. Agent benchmarks test whether a model can call tools. None of these alone measures whether an agent preserves context integrity across time.

We introduce Context Integrity Benchmark (CIB), an evaluation framework for persistent AI agents. A system has context integrity when every answer or action can be traced to the right stored evidence, updated against newer evidence, bounded by uncertainty, and executed only when the evidence supports it. CIB evaluates seven task families: selective memory writes, evidence retrieval, knowledge update, abstention, multi-session reasoning, action grounding, and causal action. CIB v0 compares recent-context retrieval, full-history context, naive lexical retrieval, write-filtered lexical retrieval, and a structured scoped-memory upper bound. We define metrics for answer accuracy, retrieval sufficiency, unsupported claim rate, stale-fact error, abstention precision and recall, action correctness, latency, token cost, and grounded utility per token.

We generate CIB v0 as a 250-task deterministic benchmark and evaluate five retrieval/memory baselines. The scoped-memory baseline reaches 100.0% retrieval sufficiency (95% Wilson CI [98.5, 100.0]) with 0.0% stale-fact error, while recent-context retrieval reaches only 16.0% sufficiency (95% CI [12.0, 21.1]). Full-history context and naive lexical retrieval both reach 100.0% evidence recall but still fail 24.0% of tasks because superseded evidence remains in context. These are not LLM agent results; they are the empirical floor for the context pipeline before answer and action models are introduced.

1. Introduction

The current AI stack has a context integrity problem.

Models can answer a question from a prompt. Retrieval systems can return a chunk. Agents can call a tool. But real work does not arrive as a single prompt with all relevant facts neatly attached. It unfolds over time. A user changes their mind. A document supersedes an older document. A preference applies in one situation but not another. An instruction is remembered, then contradicted. The agent must decide what to store, what to ignore, what to retrieve, what to update, when to ask for clarification, and whether an action is justified.

Long context is not memory. It is capacity. Memory is state maintained across time. Retrieval is not understanding. It is access. Reasoning is not a fluent explanation. It is the correct use of evidence under constraints.

Current evaluations often blur those distinctions. A model may pass a long-context needle test while still failing as a memory system because it never had to choose what to write. A RAG system may retrieve a passage while still producing unsupported claims. An agent may complete a tool-use task while relying on stale or ungrounded context. In deployment, these are not separate problems. They are the same failure viewed from different angles.

This paper proposes Context Integrity Benchmark (CIB): a benchmark for evaluating whether AI agents preserve, retrieve, update, and use context correctly across sessions.

Our main contributions are:

- We define context integrity as a measurable property of agent systems.
- We propose seven task families that stress persistent memory, retrieval, abstention, and action grounding.
- We define metrics that evaluate not only final answer accuracy but evidence use, unsupported claims, stale facts, and cost.
- We release a deterministic 250-task CIB v0 generator/evaluator with five baseline systems.
- We report initial retrieval and memory results showing that full-history context and lexical retrieval can find evidence while still failing update and causal-action tasks.
- We connect context integrity to causal action: agents that act must distinguish evidence of association from evidence that an intervention will work.

2. Definition

A system has context integrity when:

- It stores facts that matter and ignores noise.
- It retrieves the minimum sufficient evidence for the current decision.
- It updates stale facts when newer evidence supersedes older evidence.
- It preserves history when the question asks for history.
- It abstains when evidence is absent or ambiguous.
- It grounds answers and actions in retrieved evidence.
- It distinguishes observed correlations from justified interventions.

This definition is deliberately operational. Context integrity is not a vibe, a product promise, or a model capability in the abstract. It is measured by comparing a system's memory writes, retrieved sources, answers, and actions against a gold evidence graph.

2.1 Formal Task Model

A CIB task is a tuple `T = (E, q, A, G, D, y)`.

`E = {e_1, ..., e_n}` is an ordered event stream. Each event has a source ID, timestamp, text payload, scope fields such as project and domain, a write label, and optionally a supersession edge to newer evidence. `q` is the later query or decision point. `A` is the allowed action set. `G(q)` is the minimum gold evidence set required for the query. `D(q)` is the disallowed stale evidence set: sources that may be historically true but should not authorize a current answer or action. `y` is the gold answer or action.

A context pipeline consists of a writer `W`, memory state `M_t`, retriever `R`, and actor `P`. For each event, `W` decides whether and how to update `M_t`. At query time, `R(q, M_t)` returns sources `S`. The actor chooses `P(q, S) -> a`.

Retrieval is sufficient when every source in `G(q)` is contained in `S` and no source in `S` belongs to `D(q)`. An action is evidence-licensed when the selected action is entailed by sufficient evidence and not contradicted by stale evidence. This makes the benchmark model-agnostic: a language model can be weak or strong, but if the context pipeline returns insufficient or stale sources, the maximum evidence-licensed action accuracy is already bounded.

3. Related Work

Retrieval-augmented generation was introduced to address the limits of parametric memory and to give generated answers access to external knowledge (Lewis et al., 2020). RAG remains the dominant pattern for grounding LLM outputs, but RAG evaluation often focuses on retrieval relevance and answer faithfulness within a single query.

Long-context evaluation studies whether models can use information placed far inside a prompt. "Lost in the Middle" showed that language models can be sensitive to where relevant information appears in long contexts (Liu et al., 2023). These evaluations are useful, but they test a fixed input. They do not test whether a system can maintain memory across time.

MemGPT frames LLM agents as systems that require virtual context management, with memory tiers inspired by operating systems (Packer et al., 2023). LongMemEval evaluates long-term memory in chat assistants across information extraction, multi-session reasoning, temporal reasoning, knowledge updates, and abstention (Wu et al., 2024). MemoryAgentBench later formalized memory-agent evaluation around accurate retrieval, test-time learning, long-range understanding, and selective forgetting (Hu et al., 2026). Evo-Memory moves further toward streaming task settings where agents must reuse and evolve experience across task streams (Wei et al., 2026).

Production memory systems have also moved from static RAG toward extraction, consolidation, and retrieval. Mem0 compares memory-centric architectures against RAG, full-context, proprietary memory, and other memory systems on long-term conversational tasks, reporting accuracy and large latency/token reductions compared with full-context baselines (Chhikara et al., 2025). Recent survey work describes agent memory as a write-manage-read loop coupled to perception and action, and identifies contradiction handling, learned forgetting, latency budgets, and privacy governance as open engineering problems (Du, 2026).

Two recent benchmark directions are especially close to CIB. StructMemEval argues that many long-term memory benchmarks still emphasize recall and proposes testing whether agents organize long-term memory into task-appropriate structures (Shutova et al., 2026). Agentic Memory treats memory operations as tool actions learned by the policy, rather than a separate heuristic controller (Yu et al., 2026). CIB is complementary: it does not prescribe an internal memory architecture. It grades whether the resulting context state is current, scoped, auditable, sufficient, and safe to act on.

Agent benchmarks such as AgentBench and SWE-bench evaluate tool use and task completion. They are essential for measuring whether agents can act. CIB asks a prior question: whether the action is grounded in valid remembered evidence.

The Reasoning Gap benchmark from Bad Theory Labs tests whether models can distinguish observational from interventional causal queries over explicit probability tables. CIB extends this concern into agent behavior. If an agent chooses actions, it must reason about what will happen if it acts, not only what patterns appeared in the past.

4. Benchmark Design

CIB is built from multi-session workflows. Each workflow contains timestamped events, evidence sources, distractors, contradictions, user preferences, documents, possible tool actions, and final questions.

A task instance includes:

- Session events: user messages, documents, code changes, emails, calendar events, API outputs, or prior agent actions.
- Gold memory writes: the facts a memory system should store.
- Gold evidence: the source IDs required to answer or act.
- Stale evidence markers: older facts that should be superseded.
- Questions: natural-language queries over the state.
- Allowed actions: discrete actions the agent may choose.
- Gold answer and gold action.
- Split tags: task family, difficulty, domain, and failure mode.

The benchmark evaluates the full pipeline:

- Ingestion: the system receives events over time.
- Memory: the system decides what to write.
- Retrieval: the system receives a later query and retrieves evidence.
- Answer/action: the system answers or chooses an action.
- Audit: the system returns citations or source IDs.

5. Task Families

5.1 Selective Write

The system receives many events. Some are useful long-term facts. Others are noise. The benchmark checks whether the system writes durable facts without storing everything.

Example failure: the agent remembers a joke, ignores the user's invoice format preference, then fails later.

5.2 Evidence Retrieval

The system must retrieve the smallest evidence set sufficient to answer. This penalizes both missing evidence and irrelevant context flooding.

Example failure: the retriever finds the right document but also includes a contradictory old note, and the model answers from the wrong one.

5.3 Knowledge Update

The system receives a fact, then receives a newer fact that supersedes it. It must use the newer fact for current-state questions and preserve the older fact for history questions.

Example failure: the agent still uses the old API endpoint after a migration notice.

5.4 Abstention

The system must say when evidence is missing. This is central because unsupported confidence is often worse than no answer.

Example failure: the agent invents a budget number because related planning notes exist.

5.5 Multi-Session Reasoning

The answer requires combining facts from multiple sessions without pulling in unrelated context.

Example failure: the agent combines a design preference from one project with a legal constraint from another.

5.6 Action Grounding

The system chooses an action from a small action set. The action must follow from retrieved evidence.

Example failure: the agent sends a follow-up email even though the user only asked for a draft.

5.7 Causal Action

The system must distinguish observed correlation from justified intervention. This connects context integrity to causal reasoning.

Example failure: the agent recommends increasing marketing spend because revenue rose after the last campaign, even though the evidence says the revenue change was caused by seasonality.

6. Baselines

CIB v0 evaluates five retrieval and memory baselines:

- Recent3: retrieve the last three events.
- FullHistory: retrieve every event in the task history.
- Lexical3: retrieve the three events with highest word overlap against the question.
- WriteLexical3: retrieve only events marked as durable writes, then rank lexically.
- ScopedHybrid3: retrieve durable writes, prefer project/domain scope, and suppress superseded facts for current-state questions.

These baselines isolate context-system quality before answer generation. Full agent evaluations can layer an answer/action model on top of the same retrieved evidence and vary that model through a gateway such as BTL Runtime to measure model capability, cost, and latency separately from memory quality.

7. Metrics

CIB reports pipeline metrics, not only final answer accuracy.

- Answer accuracy: whether the final answer matches the gold answer.
- Action accuracy: whether the selected action matches the gold action.
- Action upper bound: the best possible action accuracy for an evidence-gated agent using only the retrieved evidence.
- Evidence recall: fraction of gold evidence source IDs retrieved.
- Evidence precision: fraction of retrieved source IDs that are gold evidence.
- Retrieval sufficiency: whether retrieved evidence is enough to answer.
- Unsupported claim rate: claims not supported by retrieved evidence.
- Stale fact error rate: answers or actions based on superseded facts.
- Unsupported-action risk: retrieved context contains no gold evidence, creating pressure for guesswork.
- Abstention precision: when the system abstains, was evidence actually missing?
- Abstention recall: when evidence was missing, did the system abstain?
- Write precision: fraction of written memories that should have been written.
- Write recall: fraction of gold memories written.
- Latency: end-to-end task time.
- Token cost: total billable input and output tokens.

The core production metric is grounded utility per token:

```
grounded_utility = supported_correct_outcomes / billable_tokens
```

This matters because production agents do not operate in a vacuum. A memory system that improves accuracy while doubling context cost may be less useful than a system that preserves groundedness under a tighter token budget.

For proportions, we report 95% Wilson confidence intervals. We use Wilson intervals rather than normal approximations because several split-level outcomes approach 0% or 100%.

For paired system comparisons, we report exact two-sided binomial tests over discordant task outcomes. Each comparison is made on the same 250 task IDs. This matters because a benchmark with heterogeneous task families can make aggregate deltas look larger or smaller depending on task mix. Paired tests ask a stricter question: on how many identical tasks did one system retrieve sufficient evidence while the other did not?

7.1 Scoring Equations

For a task with retrieved set `S`, gold evidence `G`, and stale set `D`, evidence precision is `|S intersect G| / |S|` when `S` is non-empty and 0 otherwise. Evidence recall is `|S intersect G| / |G|`. Retrieval sufficiency is 1 exactly when every gold source is retrieved and no stale source is retrieved; otherwise it is 0.

Stale error is 1 when `S` contains any source in `D`. Unsupported-action risk is 1 when `S` is non-empty but contains no source in `G`. Context flood is 1 when `|S| > |G| + 2`. The action upper bound is equal to

retrieval sufficiency in CIB v0 because a perfect actor cannot license the gold action without sufficient current evidence.

The grounded utility score used in this paper is:

```
GU_1k = 1000 * sufficient_outcomes / estimated_context_tokens
```

This is intentionally simple. Later releases should replace estimated context tokens with provider-reported billable tokens and should report latency distributions rather than averages alone.

8. CIB v0 Results

We implement CIB v0 as a deterministic 250-task synthetic benchmark. This first release evaluates retrieval and memory policies only. It does not evaluate LLM answer generation, tool execution, or frontier model behavior.

The five baselines are:

- Recent3: retrieve the last three events.
- FullHistory: retrieve every event in the task history.
- Lexical3: retrieve the three events with highest word overlap against the question.
- WriteLexical3: retrieve from events marked as durable memory writes, ranked lexically.
- ScopedHybrid3: retrieve durable writes, prefer matching project/domain scope, and suppress superseded facts for current-state questions. This is a structured-memory upper bound for CIB v0, not a claim about an existing deployed model.

System	Evidence precision	Evidence recall	Retrieval sufficiency	Action upper bound	Stale error	Unsupported risk	Avg tokens	Grounded utility / 1k tokens
recent3	18.0%	39.0%	16.0% [12.0%, 21.1%]	16.0%	8.0%	46.0%	40.0	4.00
fullHistory	29.5%	100.0%	76.0% [70.3%, 80.9%]	76.0%	24.0%	0.0%	55.5	13.68
lexical3	43.3%	100.0%	76.0% [70.3%, 80.9%]	76.0%	24.0%	0.0%	42.1	18.03
writeLexical3	88.0%	100.0%	76.0% [70.3%, 80.9%]	76.0%	24.0%	0.0%	28.0	27.10
scopedHybrid3	100.0%	100.0%	100.0% [98.5%, 100.0%]	100.0%	0.0%	0.0%	26.0	38.40

These results show five early signals. First, recency is a weak proxy for memory: the last three events often miss the evidence needed for the decision, producing a 46.0% unsupported-action risk. Second, full-history context is not enough: it reaches 100.0% recall but still fails update and causal-action tasks

because stale evidence remains available to the actor. Third, lexical retrieval can find all required evidence while still failing because it also retrieves stale confounders, producing the same 24.0% stale-error rate as full history. Fourth, explicit write filtering improves precision and token cost, but does not solve stale facts by itself. Fifth, the action upper bound tracks retrieval sufficiency exactly: if the context pipeline does not retrieve sufficient current evidence, even a perfect evidence-gated actor cannot choose the correct action. Scope and update semantics matter.

The scoped hybrid baseline is intentionally simple. It is not a claim that CIB is solved. It is a sanity-check baseline showing that the benchmark rewards systems that preserve scope, suppress superseded facts, and avoid flooding the model with irrelevant context.

8.1 Retrieval Sufficiency by Task Family

Family	recent3	fullHistory	lexical3	writeLexical3	scopedHybrid3
selective_write	0.0%	100.0%	100.0%	100.0%	100.0%
evidence_retrieval	0.0%	100.0%	100.0%	100.0%	100.0%
knowledge_update	100.0%	0.0%	0.0%	0.0%	100.0%
abstention	0.0%	100.0%	100.0%	100.0%	100.0%
multi_session	0.0%	100.0%	100.0%	100.0%	100.0%
action_grounding	0.0%	100.0%	100.0%	100.0%	100.0%
causal_action	0.0%	0.0%	0.0%	0.0%	100.0%

The split-level result clarifies the failure mode. Full-history and lexical retrieval are strong on tasks where current evidence is enough and weak precisely when the benchmark requires update semantics or causal-action discipline. They fail knowledge-update and causal-action tasks because the old evidence is present and action-relevant unless the memory system knows it has been superseded. This is the core reason context integrity cannot be reduced to context length or semantic similarity.

8.2 Paired Sufficiency Tests

Aggregate scores show the size of the gap. Paired tests show where the gap comes from. We compare each baseline against scopedHybrid3 on the same 250 tasks and count discordant outcomes: tasks where only one system retrieved sufficient, current evidence.

Baseline vs scopedHybrid3	Both sufficient	scoped only	Baseline only	Both insufficient	Delta	Exact paired p
recent3	40	210	0	0	84.0%	<0.0001
fullHistory	190	60	0	0	24.0%	<0.0001
lexical3	190	60	0	0	24.0%	<0.0001
writeLexical3	190	60	0	0	24.0%	<0.0001

The paired table makes the result harder to dismiss as an averaging artifact. ScopedHybrid3 does not trade off wins and losses against full-history context in CIB v0. It matches it on the 190 tasks where full-history is sufficient and fixes the 60 tasks where full-history exposes stale evidence. Those 60 tasks are exactly the knowledge-update and causal-action families. In other words: the measured advantage is not "better search." It is update semantics.

9. Example Task


```
{
  "id": "cib_0001",
  "sessions": [
    {
      "timestamp": "2026-06-01T09:00:00Z",
      "events": [
        {
          "source_id": "s1_e1",
          "type": "message",
          "text": "For finance exports, group invoices by client, not by month.",
          "should_write": true
        },
        {
          "source_id": "s1_e2",
          "type": "message",
          "text": "The blue dashboard mockup looked funny.",
          "should_write": false
        }
      ]
    },
    {
      "timestamp": "2026-06-12T15:30:00Z",
      "events": [
        {
          "source_id": "s2_e1",
          "type": "message",
          "text": "Actually, for audit exports only, group invoices by month.",
          "should_write": true,
          "supersedes": []
        }
      ]
    }
  ],
  "question": "How should the agent format a normal finance invoice export?",
  "gold_answer": "Group invoices by client.",
  "gold_evidence": ["s1_e1", "s2_e1"],
  "allowed_actions": ["group_by_client", "group_by_month", "ask_user"],
  "gold_action": "group_by_client",
  "requires_abstention": false,
  "split": ["preference", "multi_session", "action_grounding"]
}
```


The second event creates a narrow exception. A brittle system may treat it as a global update and group every export by month. A context-integrity-preserving system uses both evidence sources and notices the scope: audit exports changed, normal finance exports did not.

10. Falsifiable Claims

CIB makes claims that can fail.

Claim 1: Long context alone is insufficient for durable agent memory. Falsification: full-history prompting matches or beats memory systems on update, abstention, action grounding, cost, and latency.

Claim 2: Retrieval relevance is insufficient without update semantics. Falsification: relevance-only retrieval matches scoped memory on stale-fact and causal-action splits.

Claim 3: Write filtering improves precision but does not solve stale context by itself. Falsification: write-filtered retrieval matches scoped memory on update and causal-action splits.

Claim 4: Causal-action tasks expose failures not visible in recall tasks. Falsification: systems that score well on recall also score well on causal action.

11. Evaluation Protocol

CIB v0 contains 250 tasks:

- 50 selective-write tasks
- 40 evidence-retrieval tasks
- 40 knowledge-update tasks
- 35 abstention tasks
- 35 multi-session reasoning tasks
- 30 action-grounding tasks
- 20 causal-action tasks

Each task includes exact source IDs for gold evidence. Grading is mostly deterministic:

- Exact match for action labels.
- Exact source-ID comparison for retrieval.
- Rule-based stale-fact detection where possible.
- Human or LLM-assisted equivalence checks only for natural-language answer variants.

The benchmark reports aggregate results plus split-level results. A system that performs well on recall but fails abstention should not be described as having good memory.

11.1 Run Validity

A CIB run is valid only if five conditions hold. First, the system must return source IDs, not only prose citations. Second, retrieved sources must correspond to task events in the released JSONL. Third, stale evidence must remain auditable; systems may suppress stale facts at retrieval time but may not delete the stale labels from the evaluator. Fourth, model-evaluation runs must record model name, endpoint family, temperature, maximum output tokens, date, and prompt template. Fifth, any human or LLM-assisted natural-language grading must be reported separately from deterministic source-ID and action-label grading.

These rules are meant to prevent the benchmark from collapsing into prompt theater. Context integrity is an audit property. If a system cannot expose what it used, the evaluator cannot distinguish grounded memory from lucky text generation.

12. Dataset Card

Dataset name: Context Integrity Benchmark v0 (CIB-v0)

Release date: 2026-07-01

Creator: Bad Theory Labs

Size: 250 deterministic synthetic tasks, emitted as JSONL.

Task families:

- Selective write: 50 tasks
- Evidence retrieval: 40 tasks
- Knowledge update: 40 tasks
- Abstention: 35 tasks
- Multi-session reasoning: 35 tasks
- Action grounding: 30 tasks
- Causal action: 20 tasks

Data generation: Tasks are generated by `scripts/evaluate-context-integrity.mjs` from templates over project, domain, timestamp, evidence, distractor, stale-evidence, and action fields. The generator is deterministic so benchmark changes can be reviewed as code diffs.

Labels: Each task includes source-level gold evidence, stale evidence markers, an abstention flag, and a discrete gold action. The primary v0 labels are exact source IDs rather than free-form natural-language labels.

Personal data: CIB-v0 contains no personal data and no customer data. Names, projects, domains, timestamps, and documents are synthetic.

Intended use: CIB-v0 is intended for evaluating context pipelines, memory policies, retrieval sufficiency, update handling, abstention behavior, and action grounding in agent systems.

Out-of-scope use: CIB-v0 should not be used as a broad measure of general intelligence, conversational quality, world knowledge, or domain expertise. A high score on CIB-v0 does not imply an agent is safe to deploy without domain-specific evaluation.

Known limitations: The dataset is small and synthetic. Its templates make failure modes auditable but may not capture the messiness of real enterprise histories. Later versions should add human-authored tasks, longer multi-document histories, tool traces, adversarial updates, and private splits.

Reproducibility artifacts:

- Dataset: `reports/context-integrity/cib-v0-dataset.jsonl`
- Summary: `reports/context-integrity/cib-v0-summary.json`
- Report: `reports/context-integrity/cib-v0-report.md`
- Manifest: `reports/context-integrity/cib-v0-manifest.json`
- PDF: `public/context-integrity/paper.pdf`
- Appendix: `docs/context-integrity-appendix.md`
- Evaluator: `scripts/evaluate-context-integrity.mjs`
- Model harness: `scripts/run-context-integrity-model-eval.mjs`

- Validator: `scripts/validate-context-integrity-release.mjs`

The manifest records SHA-256 hashes, byte sizes, and line counts for the released dataset, reports, PDF, HTML, and scripts. `npm run cib:release` regenerates the benchmark, PDF, manifest, hash verification, and release validator in one command.

13. Frontier Model Evaluation Protocol

CIB separates context-pipeline evaluation from model evaluation. The retrieval results above answer the question: did the system surface sufficient, current, scoped evidence? Frontier model evaluation answers the next question: given the task history and evidence, does the model choose the correct action and cite the correct sources?

We provide an OpenAI-compatible evaluation harness at `scripts/run-context-integrity-model-eval.mjs`. The harness consumes `reports/context-integrity/cib-v0-dataset.jsonl`, prompts the model with timestamped events and the task question, and requires strict JSON output:

```
{
  "action": "one_of_allowed_actions",
  "evidence": ["source_id"],
  "abstain": true
}
```

The harness scores action accuracy, abstention accuracy, evidence sufficiency, evidence precision, evidence recall, and stale-evidence rate. It supports any OpenAI-compatible endpoint through `CIB_MODEL_BASE_URL`, `CIB_MODEL_API_KEY`, and `CIB_MODEL`. No frontier model scores are reported in this paper because no model credentials were available in the evaluation environment. This is a constraint on this release, not a benchmark limitation.

14. Expected Failure Modes

The benchmark is designed to surface eight common failures:

- Context flooding: retrieving too much and letting irrelevant evidence dominate.
- Quiet omission: missing the one source needed for the answer.
- Stale override: using an old fact after a newer one supersedes it.
- Over-update: treating a narrow exception as a global replacement.
- False memory: storing a summary that was never stated.
- Unsupported confidence: answering when evidence is absent.
- Action drift: choosing a plausible action not licensed by evidence.
- Causal confusion: recommending an intervention from observed correlation.

15. Discussion

The important distinction is between having context and preserving context integrity. A long prompt can contain the right sentence. A vector index can return a similar chunk. Neither guarantees that the system

knows which evidence is current, scoped, sufficient, and action-authorizing.

This is why agent memory cannot be evaluated only as recall. The practical question is not "can the system remember something?" It is "can the system maintain an auditable state that supports correct decisions over time?"

For Bad Theory Labs, this frames memory, runtime, and action as one research problem. RetainDB can be evaluated as the memory and retrieval layer. BTL Runtime can measure model, latency, and token-cost effects. Marrow can be evaluated as the action layer that must decide when evidence is strong enough to intervene.

16. Limitations

This paper reports retrieval and memory baselines, not frontier LLM agent results. The strongest agent-level claims require running answer and action models over the retrieved evidence. Synthetic tasks may encode designer bias, so later versions should include human-authored and generated variants with human review. LLM-based answer grading should be minimized because it can introduce evaluator bias. Finally, context integrity is broad; a 250-task benchmark cannot cover every domain where agents operate.

17. Threats to Validity and Governance

CIB is vulnerable to benchmark overfitting if the public synthetic templates become the training target. This is why later releases should include private splits, held-out human-authored workflows, and adversarial updates not visible in the generator. The v0 release is best understood as an auditable seed benchmark and a specification of failure modes, not a final leaderboard.

There is also a governance risk. A high CIB score means a system preserved and used benchmark evidence correctly. It does not prove that the system is safe to deploy in finance, medicine, law, security, or other high-stakes domains. Deployment requires domain-specific evidence policies, privacy controls, retention limits, human override paths, and logging for contested actions.

The benchmark intentionally requires source IDs because context integrity should be inspectable. A system that cannot explain which stored evidence authorized an answer or action should not receive credit, even if its prose answer is correct. This favors auditable systems over opaque memory claims.

18. Conclusion

The next wave of AI agents will be judged less by whether they can talk and more by whether they can follow through. Following through requires context integrity: storing the right facts, retrieving the right evidence, updating beliefs, abstaining under uncertainty, and grounding actions in what is actually known.

Context Integrity Benchmark turns that requirement into an evaluation target. If an agent cannot preserve evidence across time, it does not have memory in the sense real work requires. It has a transcript and a guess.

References

Lewis, P. et al. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. https://arxiv.org/abs/2005.11401

Liu, N. F. et al. (2023). Lost in the Middle: How Language Models Use Long Contexts. https://arxiv.org/abs/2307.03172

Packer, C. et al. (2023). MemGPT: Towards LLMs as Operating Systems. https://arxiv.org/abs/2310.08560

Wu, Z. et al. (2024). LongMemEval: Benchmarking Chat Assistants on Long-Term Interactive Memory. https://arxiv.org/abs/2410.10813

Jin, Z. et al. (2024). CLadder: Assessing Causal Reasoning in Language Models. https://arxiv.org/abs/2312.04350

Yao, S. et al. (2022). ReAct: Synergizing Reasoning and Acting in Language Models. https://arxiv.org/abs/2210.03629

Schick, T. et al. (2023). Toolformer: Language Models Can Teach Themselves to Use Tools. https://arxiv.org/abs/2302.04761

Jimenez, C. E. et al. (2023). SWE-bench: Can Language Models Resolve Real-World GitHub Issues? https://arxiv.org/abs/2310.06770

Anthropic (2024). Introducing the Model Context Protocol. https://www.anthropic.com/news/model-context-protocol

Liu, X. et al. (2023). AgentBench: Evaluating LLMs as Agents. https://arxiv.org/abs/2308.03688

Chhikara, P. et al. (2025). Mem0: Building Production-Ready AI Agents with Scalable Long-Term Memory. https://arxiv.org/abs/2504.19413

Hu, Y. et al. (2026). Evaluating Memory in LLM Agents via Incremental Multi-Turn Interactions. https://arxiv.org/abs/2507.05257

Wei, T. et al. (2026). Evo-Memory: Benchmarking LLM Agent Test-time Learning with Self-Evolving Memory. https://arxiv.org/abs/2511.20857

Du, P. (2026). Memory for Autonomous LLM Agents: Mechanisms, Evaluation, and Emerging Frontiers. https://arxiv.org/abs/2603.07670

Shutova, A. et al. (2026). Evaluating Memory Structure in LLM Agents. https://arxiv.org/abs/2602.11243

Yu, Y. et al. (2026). Agentic Memory: Learning Unified Long-Term and Short-Term Memory Management for Large Language Model Agents. https://arxiv.org/abs/2601.01885

Appendix A. Task Schema

CIB tasks are JSON objects with the following required fields:

Field	Type	Meaning
`id`	string	Stable task identifier.
`family`	string	One of the seven task-family labels.
`project`	string	Scope field used to test project isolation.
`domain`	string	Scope field used to test domain isolation.
`events`	array	Timestamped event stream.
`question`	string	Later query or decision point.
`gold_evidence`	array	Source IDs required for a sufficient answer or action.
`stale_evidence`	array	Source IDs that should not authorize current action.
`requires_abstention`	boolean	Whether the correct behavior is to ask or abstain.
`gold_action`	string	Discrete gold action label.

Each event has a `source_id`, `timestamp`, `text`, `should_write`, `project`, `domain`, `stale`, and optional `superseded_by` field. A stale event must point to the newer event that supersedes it.

Appendix B. Release Checklist

A CIB v0 release should pass all of the following checks:

- `npm run cib:release`
- `npm run cib:manifest:verify`
- `npm run cib:validate`
- `npm run build`

The release validator checks task counts, family counts, source-ID uniqueness, evidence pointers, stale supersession metadata, summary totals, paired-comparison totals, metric ranges, and manifest hashes. A release should not be compared against another run until these checks pass.

Appendix C. Model Evaluation Contract

For frontier model evaluation, the model receives timestamped task events and the task question. It must return strict JSON:

```
{
  "action": "one_of_allowed_actions",
  "evidence": ["source_id"],
  "abstain": true
}
```

The harness scores action accuracy, abstention accuracy, evidence precision, evidence recall, retrieval sufficiency, and stale-evidence rate. Runs should report model name, endpoint family, date, temperature, maximum output tokens, prompt template, and any non-deterministic grading.