LONGHARNESS: ADVANCING AGENTIC REASONING OVER 100M-TOKEN CORPORA VIA STATEFUL EVIDENCE MANAGEMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Corpus-level long-context reasoning requires agents to acquire and integrate evidence across document collections that exceed their context windows. Existing agent harnesses enable incremental access to these corpora, but acquired evidence and evolving conclusions often remain entangled in a growing interaction history. This implicit state management can lead to two coupled failure modes: unbounded exploration, where retrieval expands the interaction history without resolving decision-relevant information gaps, and evidence-state drift, where previously acquired evidence or subsequent corrections fail to inform the current conclusion. We present LONGHARNESS, a model-agnostic harness that makes the interaction between acquired evidence and evolving conclusions explicit. Its core mechanism, *Evidence–Conclusion Coevolution*, uses unresolved conclusions to guide exploration toward missing information, while newly acquired evidence strengthens or revises the conclusions that directed the search. LONGHARNESS couples this process with *Evidence Lifecycle Management*, which keeps accumulated evidence current and traceable throughout the interaction. Together, these mechanisms focus retrieval on remaining information needs and keep conclusions aligned with current evidence. We evaluate LONGHARNESS on OfficeQA Pro V1 and V2, AA-LCR, and LongBench v2 using GLM-5.3-Flash and Qwen3.8-Flash. Compared with direct reasoning using the same backbone, LONGHARNESS improves GLM-5.3-Flash exact match on OfficeQA Pro from 15.8% to 60.2% on V1 and from 4.4% to 61.1% on V2. Against state-of-the-art agent harnesses, it surpasses Claude Code by 12.0 percentage points on V1 and Codex by 18.9 points on V2. With Qwen3.8-Flash, LONGHARNESS reaches 61.7% on V1, outperforming Claude Code by 8.3 points and Codex by 12.0 points.

1 INTRODUCTION

Real-world document questions often require evidence scattered across large, evolving document collections. Corpus-level long-context reasoning requires agents to locate and integrate this evidence even when the collection exceeds their context windows. Longer windows expand the material available within a single invocation (Gemini Team, 2024; Yang et al., 2025b; Shang et al., 2025), but models remain sensitive to evidence position and distractors even when the relevant content fits (Liu et al., 2024; Hsieh et al., 2024; Ling et al., 2025; Zhuang et al., 2025a). Agents must therefore search selectively, read relevant passages, and accumulate evidence across multiple interactions. As these interactions unfold, the challenge extends from finding evidence to keeping it usable, revising it when needed, and carrying it into the final decision.

We view this process at three levels. *Evidence retrieval* (Level 1) locates relevant passages while filtering distractors. *Evidence editing* (Level 2) selects, reconciles, and revises retrieved content. *Evidence transformation* (Level 3) turns the resulting evidence into an answer. Training methods supervise retrieval trajectories and memory edits (Lin et al., 2026b; Chen et al., 2026; Wang et al., 2025; Gao & Marttinen, 2026), while agent harnesses enable adaptive document access and persistent storage (Cao et al., 2026; Chen et al., 2023; Lee et al., 2024; Zhao et al., 2024; Packer et al., 2024; Karten et al., 2026; Yu et al., 2026). These advances strengthen evidence acquisition and maintenance, yet the transition from accumulated evidence to an evolving conclusion often remains

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

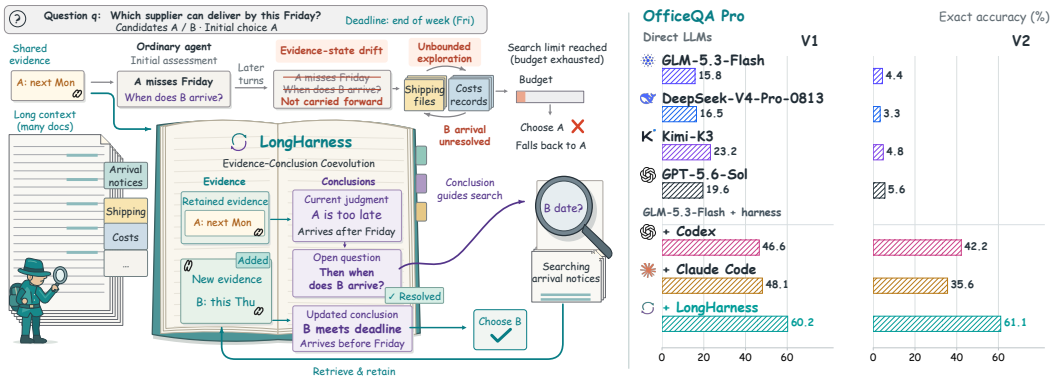


Figure 1: Evidence lifecycle management for corpus-level long-context reasoning. The left panel illustrates how an agent loses an earlier constraint and continues searching, while LONGHARNESS retains the evidence, uses an unresolved conclusion to guide retrieval, and revises its decision when new evidence arrives. The right panel compares exact accuracy (%) on OfficeQA Pro V1 and V2 for direct models and agent harnesses using GLM-5.3-Flash.

implicit. The conclusions that determine what to search for next and what answer to produce remain embedded in session history, free-form notes, or transient reasoning.

At corpus scale, this implicit transition can lead to two coupled failure modes. (i) *Unbounded exploration* occurs when retrieval continues without resolving decision-relevant information gaps, potentially exhausting the interaction budget. (ii) *Evidence-state drift* occurs when previously acquired evidence or later corrections become disconnected from subsequent reasoning, leaving the final decision based on stale or incomplete evidence. These failures can reinforce each other as drift obscures which information needs have already been resolved, prompting further retrieval that expands the history and makes relevant evidence harder to recover.

We introduce LONGHARNESS, a model-agnostic harness that maintains source-linked evidence and evolving conclusions as coupled persistent states. Unresolved conclusion gaps guide subsequent search and reading, while accepted evidence informs revisions to the conclusions and their remaining gaps. Its core mechanism, *Evidence–Conclusion Coevolution*, uses unresolved conclusions to guide exploration toward missing information, while newly acquired evidence strengthens or revises the conclusions that directed the search. By making the remaining information needs explicit, this feedback directs retrieval toward gaps that matter to the decision and helps curb unbounded exploration. LONGHARNESS couples this loop with *Evidence Lifecycle Management*, which keeps accumulated evidence current and traceable throughout the interaction. The EVIDENCEBANK retains source-linked statements and tracks which remain active throughout the evidence lifecycle, helping reduce evidence-state drift by keeping the basis for subsequent reasoning aligned with the latest findings.

We evaluate LONGHARNESS on OfficeQA Pro V1 and V2, AA-LCR, and the LongBench v2 Long subset using GLM-5.3-Flash and Qwen3.8-Flash. OfficeQA Pro requires reasoning over roughly 100M tokens in V1, while V2 doubles the corpus and introduces more demanding compositional questions. Compared with direct reasoning using the same backbone, LONGHARNESS raises GLM-5.3-Flash exact match from 15.8% to 60.2% on V1 and from 4.4% to 61.1% on V2. Against state-of-the-art agent harnesses, it surpasses Claude Code by 12.0 percentage points on V1 and Codex by 18.9 points on V2. With Qwen3.8-Flash, it reaches 61.7% on V1, outperforming Claude Code by 8.3 points and Codex by 12.0 points. Trajectory analysis with GLM-5.3-Flash further connects these gains to evidence management, with V2 success rates exceeding the stronger of Claude Code and Codex at each stage by 11.1 points in retrieval, 2.7 in evidence editing, and 9.3 in transformation. Our contributions are summarized as follows.

- We formulate corpus-level long-context reasoning as evidence lifecycle management, viewing the construction and continual refinement of the evidence supporting a decision as the central reasoning process. This perspective unifies evidence retrieval, editing, and transformation through the evolving role of evidence in decision-making.

- We develop LONGHARNESS, a model-agnostic harness that operationalizes this formulation through *Evidence–Conclusion Coevolution* and *Evidence Lifecycle Management*. It maintains evidence and conclusions as coupled persistent runtime states, allowing unresolved conclusions to direct evidence acquisition and newly acquired evidence to revise those conclusions, while an EVIDENCEBANK explicitly controls the evidence lifecycle and preserves the sources of active evidence.
- We evaluate LONGHARNESS on corpus-level reasoning over roughly 100M-token and doubled document collections, together with provided-context benchmarks, using two model backbones. With GLM-5.3-Flash, LONGHARNESS outperforms Claude Code by 12.0 percentage points on OfficeQA Pro V1 and Codex by 18.9 points on V2, while ablations and trajectory analyses reveal the contributions of conclusion-guided exploration and evidence lifecycle management.

2 METHOD

2.1 OVERVIEW

Given a question q and documents $D = \{d_1, \ldots, d_n\}$ outside the prompt, LONGHARNESS incrementally acquires evidence through an external environment $\mathcal{E}_D$ under a resource budget. Its state $S_t = (E_t, C_t, H_t)$ comprises source-linked evidence, evidence-linked conclusions and unresolved gaps, and prior reasoning, actions, and observations. Given q, document map M_D, state S_t, and remaining budget B_t, the policy selects an action and an optional conclusion update, and the environment returns the next state and observation.

$$(a_t, \widehat{C}_t) \sim \pi_\theta(\cdot \mid q, M_D, S_t, B_t), \tag{1}$$

$$(x_{t+1}, o_{t+1}) \sim \mathcal{P}_D(\cdot, \cdot \mid x_t, a_t), \tag{2}$$

Here, x_t denotes the environment state, including registered source spans, evidence entries and their lifecycle status, and resource counters. Its active evidence forms E_t in the policy state S_t. The action a_t specifies search, bounded reading, EVIDENCEBANK operations, computation, or final submission, and $\widehat{C}_t$ is the optional conclusion update. The controller validates action structure, registered source references, and budget constraints before exposing updates to subsequent turns.

LONGHARNESS couples two mechanisms (Figure 2). Evidence–Conclusion Coevolution uses unresolved conclusions to guide evidence acquisition and accepted evidence to strengthen or revise conclusions (Section 2.2). Evidence Lifecycle Management controls evidence admission, revision, retraction, and supersession while preserving source references (Section 2.3). A shared interface supports adaptive interleaving of these operations with document access, computation, and final submission (Section 2.4).

2.2 EVIDENCE–CONCLUSION COEVOLUTION

Separating evidence from inference. The EVIDENCEBANK maintains source-linked statements extracted from passages the agent has read and serves as the persistent state in which evidence is admitted and revised, while the CONCLUSION STATE records the agent's current interpretation of that evidence through established claims, supported hypotheses, and unresolved gaps. Maintaining these representations separately distinguishes the acquired evidence from the inferences drawn from it, while explicit references connect each claim to its supporting evidence. Each unresolved gap identifies information needed to establish, qualify, or revise the current conclusion. At the level of a reasoning cycle, the two states coevolve through conclusion-guided evidence acquisition and evidence-grounded conclusion revision,

$$E^+ = \mathcal{A}_\theta(q, D, E, C), \tag{3}$$

$$C^+ = \mathcal{R}_\theta(q, E^+, C). \tag{4}$$

where $\mathcal{A}_\theta$ represents exploration guided by the current conclusions and admission of the resulting statements into the EVIDENCEBANK, while $\mathcal{R}_\theta$ updates the CONCLUSION STATE using the active evidence in the EVIDENCEBANK. The same agent policy realizes both operations over one or more interactions, with the equations specifying their dependencies while allowing exploration and revision to be interleaved adaptively.

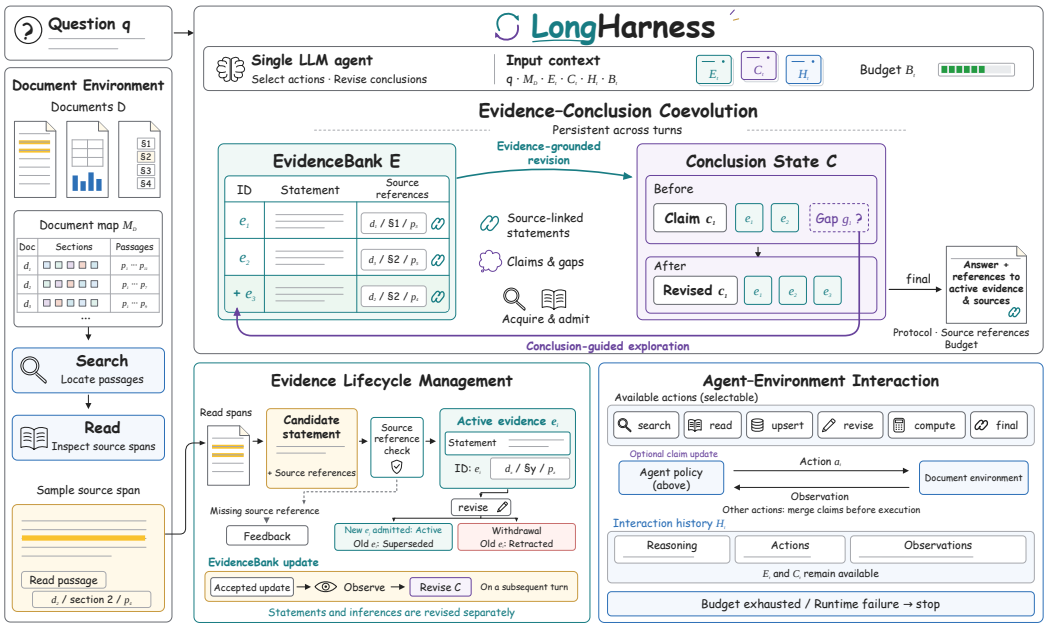


Figure 2: Overview of LONGHARNESS. A single agent maintains an EVIDENCEBANK and a CONCLUSION STATE while interacting with external documents. Evidence–Conclusion Coevolution uses unresolved conclusions to guide exploration and accepted evidence to revise those conclusions, while Evidence Lifecycle Management governs which statements remain active and preserves their source references throughout the interaction.

Conclusion-guided exploration. The current CONCLUSION STATE guides what the agent searches for and reads next. For each claim, the agent identifies observations that would establish, qualify, or refute it and uses these requirements to formulate queries and select reading scopes. Competing hypotheses motivate searches that distinguish between them, while unresolved gaps direct attention to missing premises. When an observation leaves a gap unresolved, the agent can reformulate the query or broaden the reading scope while keeping further exploration tied to the information needed to advance the conclusion.

Evidence-grounded revision. After observing the result of an EVIDENCEBANK update, the agent reassesses the claims in the CONCLUSION STATE against the bank's active evidence and determines whether they remain supported, require qualification, or must be revised. It updates the affected claims together with their evidence references and unresolved gaps, keeping the CONCLUSION STATE aligned with the current evidence. Before final submission, the agent incorporates any relevant evidence not yet reflected in this state and checks whether the remaining gaps could change the requested decision.

2.3 EVIDENCE LIFECYCLE MANAGEMENT

Evidence Lifecycle Management governs how source-linked statements enter, persist in, and leave the active evidence state in the EVIDENCEBANK. A statement begins as a candidate extracted from a passage the agent has read, becomes active upon admission, and may subsequently be revised or retired as new evidence arrives. This lifecycle requires (i) traceability to observed source passages, (ii) persistent availability across interactions, and (iii) explicit updates when a statement's supporting evidence changes.

The EVIDENCEBANK implements these requirements through source-linked admission, persistent storage, and explicit revision. For traceability, each candidate statement must cite source references registered by prior read actions, with the controller checking reference validity and the agent judging whether the statement faithfully represents its sources. For persistence, admitted statements remain available in the EVIDENCEBANK across subsequent interactions, allowing the agent to reuse them without reconstructing the evidence from earlier observations. For explicit revision, a corrected

statement is admitted as a new active entry and the previous entry is marked as superseded, while a retracted statement leaves the active state without a replacement. The agent observes these EVIDENCEBANK updates before revising the affected conclusions, and final references resolve to active statements and their underlying source passages.

2.4 AGENT–ENVIRONMENT INTERACTION

At each turn, the policy selects an action a_t and an optional conclusion update $\widehat{C}_t$ under the formulation in Section 2.1. The environment executes the action and returns an observation, while the harness enforces the interaction protocol, source-reference constraints, and resource limits. The shared action space lets the policy interleave evidence acquisition, maintenance, and use. *Search* locates candidate documents or passages, and *Read* returns selected text and registers source references. The agent proposes source-linked evidence through `upsert` and corrects or retracts entries through `revise` (Section 2.3). *Compute* evaluates numerical expressions supplied by the policy and validates their declared evidence references, and *Final* submits the answer with supporting references.

EVIDENCEBANK mutations defer conclusion updates until the agent observes their results. The policy then revises affected claims and references through optional partial updates, with omitted fields retained. Superseding or retracting evidence does not automatically rewrite dependent conclusions. The controller checks declared evidence references at computation and final submission, while the policy remains responsible for source fidelity and whether the evidence supports its conclusions. Reasoning, actions, and observations accumulate in the history, while evidence and conclusions persist across turns. Final acceptance requires a valid answer structure and references resolving to active evidence and source passages. Resource exhaustion or runtime failure may terminate the episode without an accepted answer. Algorithm 1 in Appendix B.5 summarizes the procedure.

3 EXPERIMENTS

3.1 EXPERIMENTAL SETUP

Benchmarks. We evaluate on OfficeQA Pro V1 and V2 (Opsahl-Ong et al., 2026), AA-LCR (Artificial Analysis Team, 2026), and the Long subset of LongBench v2 (Bai et al., 2024), covering both corpus-level reasoning and reasoning over provided contexts. OfficeQA Pro V1 contains 697 Treasury documents totaling approximately 100M tokens, while V2 expands the corpus to 1,435 documents and introduces composite statistical questions that require combining values from multiple sources. Agents receive access to the full corpus without question-specific gold documents, making source discovery part of the task. AA-LCR provides the relevant context for each of its 100 questions, while the LongBench v2 Long subset evaluates reasoning over the benchmark's longest contexts. We report exact match and accuracy within 5% relative error on OfficeQA Pro, and Exact accuracy on AA-LCR and LongBench v2.

Models and harnesses. Our direct-reasoning baselines include Qwen3.8-27B, Qwen3.7-Max, and Qwen3.8-Flash (Yang et al., 2025a); DeepSeek-V4-Flash-0731 and DeepSeek-V4-Pro-0813 (DeepSeek-AI et al., 2026); Gemini-3.8-Flash (Google DeepMind, 2026); GPT-5.6-Luna and GPT-5.6-Sol (Singh et al., 2026); GLM-5.3 and GLM-5.3-Flash (GLM-5-Team et al., 2026); and Kimi-K3 (Kimi Team et al., 2026). Citations refer to the corresponding model families or release-specific model cards. On OfficeQA Pro, these models receive question-relevant passages retrieved with BM25 into a 256K-token context window, while LongBench v2 contexts follow the official truncation procedure with the same token limit. We compare Codex (OpenAI, 2026), Claude Code (Anthropic, 2026), RLM (Zhang et al., 2025), and LONGHARNESS using GLM-5.3-Flash and Qwen3.8-Flash as matched backbones, holding model identity fixed within each comparison. All harnesses receive the same questions and document corpus without access to reference answers or gold source annotations.

Implementation details. All backbone models use temperature 1, top-p 0.95, maximum reasoning effort, and an output limit of 32,768 tokens per request. Codex and Claude Code each allow up to 200 model turns and 200 tool calls, while RLM allows 30 iterations and LONGHARNESS allows

Table 1: Main results (%) across four benchmarks. OfficeQA Pro reports exact match (Ex.) and accuracy within 5% relative error (@5%), while AA-LCR and the LongBench v2 Long subset report Exact accuracy. Direct baselines use BM25 retrieval into a 256K-token window on OfficeQA Pro. "-" denotes direct reasoning without a harness, and "Avg." averages the Exact scores across benchmarks. Each harness result is from one run per setting. Direct AA-LCR scores are reported by Artificial Analysis; harness AA-LCR scores are evaluated in this work.

Model	Harness	OfficeQA V1		OfficeQA V2		AA-LCR	LongBench V2	Avg.
		Ex.	@5%	Ex.	@5%			
Qwen3.8-27B	-	15.8	27.1	4.4	13.3	82.0	63.0	41.3
DeepSeek-V4-Flash-0731	-	14.3	27.1	3.3	6.7	79.7	53.7	37.8
Gemini-3.8-Flash	-	18.1	27.8	4.4	13.3	81.3	70.4	43.5
GPT-5.6-Luna	-	17.3	37.6	2.2	13.3	83.7	61.1	41.1
GLM-5.3	-	12.8	18.8	3.3	13.3	79.7	60.2	39.0
Qwen3.7-Max	-	13.5	27.8	3.3	22.2	79.0	64.8	40.2
Kimi-K3	-	23.2	34.7	4.8	16.1	88.7	68.0	46.2
DeepSeek-V4-Pro-0813	-	16.5	30.1	3.3	16.7	80.3	63.5	40.9
GPT-5.6-Sol	-	19.6	44.4	5.6	33.3	84.0	64.8	43.5
GLM-5.3-Flash	-	15.8	27.8	4.4	15.6	80.0	62.0	40.6
GLM-5.3-Flash	Codex	46.6	61.7	42.2	62.2	77.0	69.4	58.8
GLM-5.3-Flash	Claude Code	48.1	67.7	35.6	58.9	79.0	63.0	56.4
GLM-5.3-Flash	RLM	45.1	62.4	38.9	58.9	56.0	60.2	50.0
GLM-5.3-Flash	**LONGHARNESS**	**60.2**	**76.7**	**61.1**	**74.4**	**84.0**	**75.9**	**70.3**
Qwen3.8-Flash	-	15.8	28.6	4.4	15.6	80.0	54.3	38.6
Qwen3.8-Flash	Codex	49.6	69.2	50.0	70.0	81.0	69.4	62.5
Qwen3.8-Flash	Claude Code	53.4	72.9	41.1	52.2	83.0	63.9	60.3
Qwen3.8-Flash	RLM	48.1	65.4	35.6	48.9	73.0	69.4	56.5
Qwen3.8-Flash	**LONGHARNESS**	**61.7**	**75.9**	**51.1**	**71.1**	**87.0**	**76.9**	**69.2**

200 steps subject to a cumulative read budget of 1,000,000 tokens and a limit of 60 search calls. We grade AA-LCR harness responses with GPT-5.6-Luna. Due to computational resource constraints, the main comparison uses one run per harness and evaluation setting. To assess variability across runs, Appendix D reports three-run results with GLM-5.3-Flash, including individual scores, means, standard deviations, and maxima. Missing answers and agent termination failures are counted as incorrect.

3.2 MAIN RESULTS

Table 1 shows that direct baselines reach at most 23.2% exact match on OfficeQA Pro V1 and 5.6% on V2 despite receiving question-relevant passages in a 256K-token window. At corpus scale, this fixed window cannot accommodate all potentially relevant evidence, and passages selected before reasoning may omit information needed to resolve intermediate conclusions. LONGHARNESS uses these conclusions to guide further retrieval and accumulates the resulting evidence across interactions, raising GLM-5.3-Flash exact match from 15.8% to 60.2% on V1 and from 4.4% to 61.1% on V2.

With GLM-5.3-Flash, LONGHARNESS outperforms the strongest competing harness by 12.0 percentage points on OfficeQA Pro V1 and 18.9 points on V2. The larger advantage on V2 is consistent with maintaining evidence across interactions as more scattered values must be located, reconciled, and combined. With Qwen3.8-Flash, LONGHARNESS also leads on both versions, reaching 61.7% on V1 and 51.1% on V2. Its advantage over the strongest competing harness is 8.3 points on V1 but narrows to 1.1 points over Codex on V2, showing that the size of the gain varies with the backbone and evaluation setting. On the LongBench v2 Long subset, LONGHARNESS reaches 75.9% with GLM-5.3-Flash and 76.9% with Qwen3.8-Flash, exceeding the next-best harness by 6.5 and 7.4 percentage points, respectively. On AA-LCR, it scores 84.0% and 87.0%, leading the strongest competing harness by 5.0 and 4.0 points. LONGHARNESS achieves the highest average Exact score across the four evaluation settings with both backbones, reaching 70.3% and 69.2%, respectively.

Table 2: Ablation results with GLM-5.3-Flash on OfficeQA Pro V1 and V2. Ex. denotes exact-match accuracy (%), and Δ denotes the decrease from full LONGHARNESS in percentage points, computed before rounding.

Variant	OfficeQA Pro V1		OfficeQA Pro V2	
	Ex.	Δ	**Ex.**	Δ
LONGHARNESS (full)	**60.2**	–	**61.1**	–
w/o Persistent States	55.6	4.5	53.3	7.8
w/o Evidence–Conclusion Coevolution	54.9	5.3	51.7	9.4
Naive Evidence Accumulation	56.4	3.8	51.1	10.0
w/o Explicit Conclusion-Guided Exploration	55.6	4.5	53.3	7.8
Generic Working Notes	53.4	6.8	53.3	7.8

Table 3: Success rates (%) for evidence retrieval, editing, and transformation on OfficeQA Pro V1 and V2 with GLM-5.3-Flash. Retrieval measures whether the required evidence is found, editing measures whether retrieved evidence is retained and adopted in the final decision, and transformation measures whether the adopted evidence yields an exactly correct answer. Editing and transformation rates are calculated only over cases that pass the preceding stages. Each harness is evaluated on one run per split.

Evidence level	OfficeQA Pro V1			OfficeQA Pro V2		
	Claude Code	Codex	LONGHARNESS	Claude Code	Codex	LONGHARNESS
L1: Retrieval (r_1)	**85.0**	72.2	**85.0**	72.2	76.7	**87.8**
L2: Evidence editing (r_2)	85.8	83.3	**87.6**	84.6	78.3	**87.3**
L3: Transformation (r_3)	66.0	77.5	**80.8**	58.2	70.4	**79.7**

3.3 ABLATION STUDY

We ask whether evidence–conclusion coevolution and evidence lifecycle management are critical to the effectiveness of LONGHARNESS, and examine this question through five ablations in Table 2. **(i) w/o Persistent States** removes the EVIDENCEBANK and CONCLUSION STATE while retaining interaction history. **(ii) w/o Evidence–Conclusion Coevolution** retains the EVIDENCEBANK but removes the CONCLUSION STATE and its retrieval feedback. **(iii) Naive Evidence Accumulation** preserves conclusion feedback but permits only append-only evidence updates. **(iv) w/o Explicit Conclusion-Guided Exploration** retains both states but removes the instructions to guide retrieval using conclusion gaps. **(v) Generic Working Notes** retains the EVIDENCEBANK and replaces structured conclusions with editable free-text notes that can record conclusions, unresolved questions, and evidence references.

Removing both persistent states reduces exact match by 4.5 percentage points on V1 and 7.8 on V2, showing that explicit evidence and conclusion states contribute beyond access to the interaction history. Removing the maintained CONCLUSION STATE and its retrieval feedback reduces accuracy by 5.3 and 9.4 points. Restricting the EVIDENCEBANK to append-only accumulation yields 56.4% on V1 and 51.1% on V2, corresponding to decreases of 3.8 and 10.0 points. These results support the value of revising and retiring evidence as new observations arrive, even when conclusion feedback is preserved. Removing the explicit conclusion-guided exploration instructions while retaining both persistent states reduces accuracy by 4.5 and 7.8 points, supporting the value of instructing the policy to use conclusion gaps to guide retrieval beyond making those states available. Replacing structured conclusions with generic working notes reduces accuracy by 6.8 and 7.8 points, showing that the full design outperforms editable free-text notes that can also retain conclusions, open questions, and evidence references. All five ablations incur larger losses on V2, consistent with the greater importance of maintaining and reconciling evidence when questions require combining values from multiple sources.

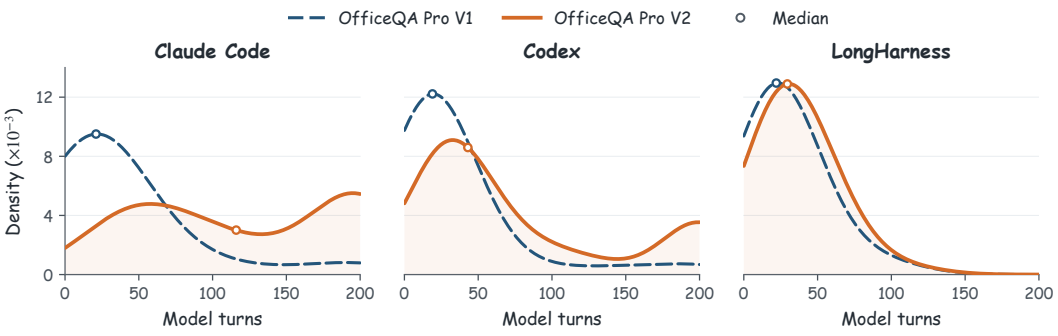


Figure 3: Distributions of model turns per question on OfficeQA Pro V1 and V2 with GLM-5.3-Flash. Each panel compares the two versions for one harness. Curves show kernel density estimates, and circles indicate medians.

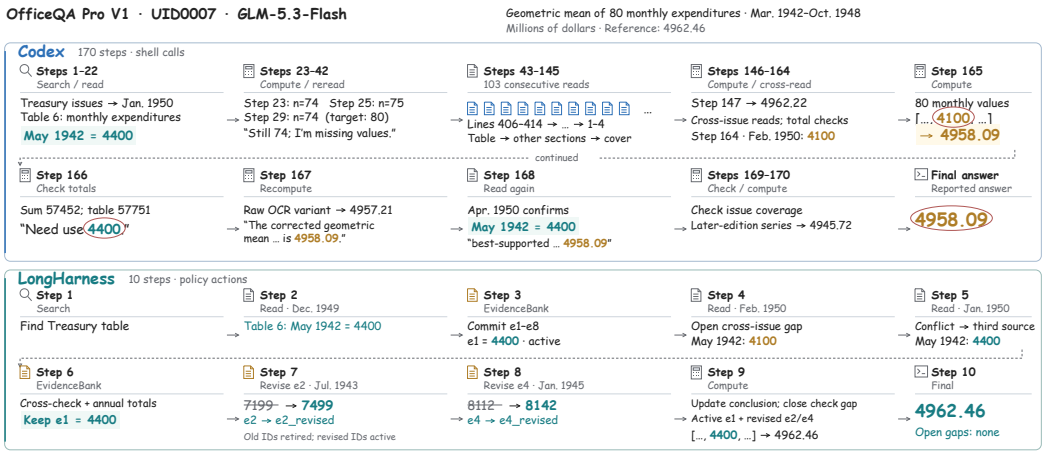


Figure 4: Evidence lifecycle comparison with GLM-5.3-Flash on OfficeQA Pro V1. For the geometric mean of 80 monthly Treasury expenditures, Codex continues searching and omits a recognized correction; LONGHARNESS updates its evidence and returns the reference answer, 4962.46 million dollars. Circles mark Codex's incorrect input, correction, and final answer.

3.4 INTERACTION LENGTH AND TRAJECTORY ANALYSIS

We ask where agent harnesses fail across evidence retrieval, editing, and transformation, and how these failures change from OfficeQA Pro V1 to V2. For each harness, we use GPT-5.6-Luna at temperature 0 to audit incorrect episodes and assess evidence retrieval, evidence adoption, and answer transformation. Exact-correct episodes are counted as passing the first two stages. Let E_1 denote the presence of required evidence in the trajectory, E_2 its retention and reconciliation for use in the final decision, and E_3 an exactly correct answer. We measure retrieval coverage and conditional editing and transformation success as

$$r_1 = P(E_1), \qquad r_2 = P(E_2 \mid E_1), \qquad r_3 = P(E_3 \mid E_1, E_2). \tag{5}$$

Table 3 shows that LONGHARNESS matches Claude Code's retrieval coverage on V1 and achieves higher conditional evidence-adoption and answer-conversion rates. On V2, its retrieval, evidence-adoption, and answer-conversion rates are 87.8%, 87.3%, and 79.7%, respectively. The conditional adoption and conversion rates remain close to their V1 values of 87.6% and 80.8%, indicating similar evidence-use outcomes across the two evaluated settings. The conditional rates are computed over the questions that pass the preceding stages for each harness and describe stage outcomes within the analyzed runs.

Figure 3 shows that median turns increase from 22 to 30 for LONGHARNESS, from 21 to 116 for Claude Code, and from 19 to 43 for Codex between V1 and V2. Codex exhausts its recorded

model-turn budget on 3.8% of V1 questions and 21.1% of V2 questions. Claude Code's distribution also concentrates more heavily near 200 turns on V2. V2 combines a larger corpus with questions requiring values from multiple sources. LONGHARNESS records unresolved information needs in the CONCLUSION STATE and retains acquired evidence in the EVIDENCEBANK, allowing subsequent retrieval to focus on missing information while reusing values already established. As questions require more cross-source checks, this organization helps avoid repeatedly searching for or re-establishing the same facts. This offers a plausible explanation for the smaller shift in LONGHARNESS's turn distribution from V1 to V2. Figure 4 compares how Codex and LONGHARNESS carry evidence corrections into the final computation, with a detailed case analysis in Appendix E.

4 RELATED WORK

4.1 LONG-CONTEXT REASONING

Post-training supervises evidence acquisition and use. LongRLVR rewards correct identification of evidence chunks (Chen et al., 2026), LongPAS assigns advantages to reasoning steps (Peng et al., 2026), and LongR measures document consultation through information gain (Ping et al., 2026). Training data include key-fact chains (Wang et al., 2025), distractor-augmented search trajectories (Lin et al., 2026b), capability-aligned task mixtures (Lv et al., 2026), and verifiable multi-hop or memory-management tasks (Wan et al., 2025; Shen et al., 2025). Training also uses agent-generated data and supervision for memory operations. ACC compiles agent trajectories into long-context QA training examples (Su et al., 2026), AgenticLU distills self-generated clarification trajectories without external teachers (Zhuang et al., 2025b), MAVEN rewards evidence-memory edits (Gao & Marttinen, 2026), and Memory-R2 credits individual memory operations (Yan et al., 2026). LONGHARNESS instead couples evidence and conclusions at runtime, linking retrieval, editing, and transformation without additional policy training.

4.2 AGENTIC HARNESSES

ReAct couples reasoning with action (Yao et al., 2023), recursive language models process context under program control (Zhang et al., 2025), and Prime Agent maintains persistent REPL state (Karten et al., 2026). Codex and Claude Code support long-context reasoning (Cao et al., 2026); other harnesses develop bounded long-horizon decomposition, harness maintenance, and persistent memory (Zheng et al., 2026; Lin et al., 2026a; Packer et al., 2024; Yu et al., 2026; Zhang et al., 2026; Pan et al., 2026). MemWalker uses memory trees (Chen et al., 2023), ReadAgent uses gist representations (Lee et al., 2024), GraphReader uses chunk graphs (Li et al., 2024), and multi-agent systems coordinate readers (Zhao et al., 2024; Zhang et al., 2024). SLIDERS reconciles evidence under an induced schema (Joshi et al., 2026), ReContext replays found-but-unused evidence (Zhao et al., 2026), and TRACE evaluates tool-augmented reasoning trajectories (Kim et al., 2026). LedgerMind uses grounding checks and event-triggered repair to address unsupported or inconsistent claims (Du et al., 2026). However, existing claims can remain grounded while the evidence needed to answer the question is still incomplete. In contrast, LONGHARNESS focuses on corpus-level long-context reasoning, maintaining source-linked evidence alongside evolving conclusions and unresolved information needs. These needs guide subsequent search and reading, while newly acquired evidence informs revisions to conclusions and remaining gaps.

5 CONCLUSION

We propose LONGHARNESS, a model-agnostic harness that unifies evidence retrieval, editing, and transformation for corpus-level long-context reasoning. Evidence–Conclusion Coevolution couples exploration with conclusion revision, while Evidence Lifecycle Management keeps evidence current and traceable. Evaluations on OfficeQA Pro, AA-LCR, and LongBench v2 show gains with two backbones. With GLM-5.3-Flash, LONGHARNESS exceeds Claude Code by 12.0 percentage points on OfficeQA Pro V1 and Codex by 18.9 points on V2. Ablations support the complementary roles of persistent states, conclusion-guided retrieval, and evidence revision. Trajectory analysis shows similar conditional evidence-use rates across V1 and V2 and a smaller increase in interaction length for LONGHARNESS than for the competing harnesses.

AI USE STATEMENT

We used AI tools to assist with language polishing and the visual presentation of the manuscript. For the text, this assistance focused on improving clarity, fluency, and consistency of expression. For figures and tables, AI tools assisted with stylistic refinement, including layout, typography, color choices, and formatting consistency. These uses supported the presentation and communication of the research.

REFERENCES

Anthropic. Claude Code. Software, 2026. URL https://github.com/anthropics/claude-code.

Artificial Analysis Team. Artificial analysis long context reasoning (AA-LCR) dataset. Hugging Face Datasets, 2026. URL https://huggingface.co/datasets/ArtificialAnalysis/AA-LCR.

Yushi Bai, Shangqing Tu, Jiajie Zhang, Hao Peng, Xiaozhi Wang, Xin Lv, Shulin Cao, Jiazheng Xu, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. LongBench v2: Towards deeper understanding and reasoning on realistic long-context multitasks. *arXiv preprint arXiv:2412.15204*, 2024. URL https://arxiv.org/abs/2412.15204.

Weili Cao, Xunjian Yin, Bhuwan Dhingra, and Shuyan Zhou. Coding agents are effective long-context processors. *arXiv preprint arXiv:2603.20432*, 2026. URL https://arxiv.org/abs/2603.20432.

Guanzheng Chen, Michael Qizhe Shieh, and Lidong Bing. Longrlvr: Long-context reinforcement learning requires verifiable context rewards. *arXiv preprint arXiv:2603.02146*, 2026. URL https://arxiv.org/abs/2603.02146.

Howard Chen, Ramakanth Pasunuru, Jason Weston, and Asli Celikyilmaz. Walking down the memory maze: Beyond context limit through interactive reading. *arXiv preprint arXiv:2310.05029*, 2023. URL https://arxiv.org/abs/2310.05029.

DeepSeek-AI et al. DeepSeek-V4: Towards Highly Efficient Million-Token Context Intelligence, 2026. URL https://arxiv.org/abs/2606.19348.

Enjun Du, Hange Zhou, Chenxu Du, Siyi Liu, Zirong Chen, Ziyu Zheng, and Yongqi Zhang. LEDGERMIND: Provenance-constrained multimodal agentic reasoning with a structured evidence ledger. *arXiv preprint arXiv:2607.28374*, 2026. URL https://arxiv.org/abs/2607.28374.

Ya Gao and Pekka Marttinen. Evidence-state rewards for long-context reasoning. *arXiv preprint arXiv:2607.02073*, 2026. URL https://arxiv.org/abs/2607.02073.

Gemini Team. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024. URL https://arxiv.org/abs/2403.05530.

GLM-5-Team et al. GLM-5: from Vibe Coding to Agentic Engineering, 2026. URL https://arxiv.org/abs/2602.15763.

Google DeepMind. Gemini 3.8 Flash Model Card, September 2026. URL https://deepmind.google/models/model-cards/gemini-3-8-flash/.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. RULER: What's the real context size of your long-context language models? *Proceedings of the Conference on Language Modeling*, 2024. URL https://arxiv.org/abs/2404.06654.

Harshit Joshi, Priyank Shethia, Jadelynn Dao, and Monica S. Lam. SLIDERS: Systematic reviews via automated evidence synthesis and reconciliation. *arXiv preprint arXiv:2604.22294*, 2026. URL https://arxiv.org/abs/2604.22294.

Seth Karten, Alex L. Zhang, Kevin Thomas, Sebastian Mueller, Elie Bakouch, Daniel Auras, Mika Senghaas, Fares Obeid, Konstantin Dunas, Johannes Hagemann, and Sami Jaghouar. Prime agent: A self-improving rlm harness. *arXiv preprint arXiv:2608.23552*, 2026. URL https://arxiv.org/abs/2608.23552.

Wonjoong Kim, Sangwu Park, Yeonjun In, Sein Kim, Dongha Lee, and Chanyoung Park. Beyond the final answer: Evaluating the reasoning trajectories of tool-augmented agents. *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2510.02837.

Kimi Team et al. Kimi K3: Open Frontier Intelligence, 2026. URL https://arxiv.org/abs/2607.24653.

Kuang-Huei Lee, Xinyun Chen, Hiroki Furuta, John Canny, and Ian Fischer. A human-inspired reading agent with gist memory of very long contexts. *arXiv preprint arXiv:2402.09727*, 2024. URL https://arxiv.org/abs/2402.09727.

Shilong Li, Yancheng He, Hangyu Guo, Xingyuan Bu, Ge Bai, Jie Liu, Jiaheng Liu, Xingwei Qu, Yangguang Li, Wanli Ouyang, Wenbo Su, and Bo Zheng. GraphReader: Building graph-based agent to enhance long-context abilities of large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 12758–12786, 2024. URL https://aclanthology.org/2024.findings-emnlp.746/.

Jiahang Lin, Shichun Liu, Chengjun Pan, Lizhi Lin, Shihan Dou, Zhiheng Xi, Xuanjing Huang, Hang Yan, Zhenhua Han, Tao Gui, and Yu-Gang Jiang. Agentic harness engineering: Observability-driven automatic evolution of coding-agent harnesses. *arXiv preprint arXiv:2604.25850*, 2026a. URL https://arxiv.org/abs/2604.25850.

Nianyi Lin, Jiajie Zhang, Lei Hou, and Juanzi Li. Longtracerl: Learning long-context reasoning from search agent trajectories with rubric rewards. *arXiv preprint arXiv:2605.31584*, 2026b. URL https://arxiv.org/abs/2605.31584.

Zhan Ling, Kang Liu, Kai Yan, Yifan Yang, Weijian Lin, Ting-Han Fan, Lingfeng Shen, Zhengyin Du, and Jiecao Chen. Longreason: A synthetic long-context reasoning benchmark via context expansion. *arXiv preprint arXiv:2501.15089*, 2025. URL https://arxiv.org/abs/2501.15089.

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. URL https://aclanthology.org/2024.tacl-1.9/.

Minxuan Lv, Tiehua Mei, Tanlong Du, Junmin Chen, Zhenpeng Su, Ziyang Chen, Ziqi Wang, Zhennan Wu, Ruotong Pan, Jian Liang, Ruiming Tang, and Han Li. GoLongRL: Capability-oriented long context reinforcement learning with multitask alignment. *arXiv preprint arXiv:2605.19577*, 2026. URL https://arxiv.org/abs/2605.19577.

OpenAI. Codex CLI. Software, 2026. URL https://github.com/openai/codex.

Krista Opsahl-Ong, Arnav Singhvi, Jasmine Collins, Ivan Zhou, Cindy Wang, Ashutosh Baheti, Owen Oertell, Jacob Portes, Sam Havens, Erich Elsen, Michael Bendersky, Matei Zaharia, and Xing Chen. OfficeQA Pro: An enterprise benchmark for end-to-end grounded reasoning, 2026. URL https://arxiv.org/abs/2603.08655.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. Memgpt: Towards llms as operating systems, 2024. URL https://arxiv.org/abs/2310.08560.

Zhuoshi Pan, Qizhi Pei, Junru Lu, Honglin Lin, H. Vicky Zhao, Di Yin, and Xing Sun. ContextPilot: Teaching agents for proactive context management via fine-grained RL. In *Proceedings of the 2026 Conference on Empirical Methods in Natural Language Processing*, 2026. URL https://arxiv.org/abs/2608.28476.

Miao Peng, Weizhou Shen, Nuo Chen, Chenliang Li, Ming Yan, and Jia Li. Incentivizing in-depth reasoning over long contexts with process advantage shaping. *arXiv preprint arXiv:2601.12465*, 2026. URL https://arxiv.org/abs/2601.12465.

Bowen Ping, Zijun Chen, Yiyao Yu, Tingfeng Hui, Junchi Yan, and Baobao Chang. LongR: Unleashing long-context reasoning via reinforcement learning with dense utility rewards. *arXiv preprint arXiv:2602.05758*, 2026. URL https://arxiv.org/abs/2602.05758.

Ning Shang, Li Lyna Zhang, Siyuan Wang, Gaokai Zhang, Gilsinia Lopez, Fan Yang, Weizhu Chen, and Mao Yang. LongRoPE2: Near-lossless LLM context window scaling. *arXiv preprint arXiv:2502.20082*, 2025. URL https://arxiv.org/abs/2502.20082.

Weizhou Shen, Ziyi Yang, Chenliang Li, Zhiyuan Lu, Miao Peng, Huashan Sun, Yingcheng Shi, Shengyi Liao, Shaopeng Lai, Bo Zhang, Dayiheng Liu, Fei Huang, Jingren Zhou, and Ming Yan. QwenLong-L1.5: Post-training recipe for long-context reasoning and memory management. *arXiv preprint arXiv:2512.12967*, 2025. URL https://arxiv.org/abs/2512.12967.

Aaditya Singh, Adam Fry, Adam Perelman, et al. OpenAI GPT-5 System Card, 2026. URL https://arxiv.org/abs/2601.03267.

Qisheng Su, Zhen Fang, Shiting Huang, Yu Zeng, Yiming Zhao, Kou Shi, Ziao Zhang, Lin Chen, Zehui Chen, Lijun Wu, and Feng Zhao. Acc: Compiling agent trajectories for long-context training. *arXiv preprint arXiv:2605.21850*, 2026. URL https://arxiv.org/abs/2605.21850.

Fanqi Wan, Weizhou Shen, Shengyi Liao, Yingcheng Shi, Chenliang Li, Ziyi Yang, Ji Zhang, Fei Huang, Jingren Zhou, and Ming Yan. Qwenlong-l1: Towards long-context large reasoning models with reinforcement learning, 2025. URL https://arxiv.org/abs/2505.17667.

Siyuan Wang, Gaokai Zhang, Li Lyna Zhang, Ning Shang, Fan Yang, Dongyao Chen, and Mao Yang. Loongrl: Reinforcement learning for advanced reasoning over long contexts. *arXiv preprint arXiv:2510.19363*, 2025. URL https://arxiv.org/abs/2510.19363.

Sikuan Yan, Ahmed Bahloul, Ercong Nie, Susanna Schwarzmann, Riccardo Trivisonno, Volker Tresp, and Yunpu Ma. Memory-R2: Fair credit assignment for long-horizon memory-augmented LLM agents. *arXiv preprint arXiv:2605.21768*, 2026. URL https://arxiv.org/abs/2605.21768.

An Yang, Anfeng Li, Baosong Yang, et al. Qwen3 Technical Report, 2025a. URL https://arxiv.org/abs/2505.09388.

An Yang, Bowen Yu, Chengyuan Li, et al. Qwen2.5-1M technical report. *arXiv preprint arXiv:2501.15383*, 2025b. URL https://arxiv.org/abs/2501.15383.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=WE_vluYUL-X.

Zhaochen Yu, Yingcheng Wu, Zhenfei Yin, Kaiyuan Chen, Zhe Zhao, Mengdi Wang, Shuicheng Yan, and Ling Yang. Recursive experiential–working memory evolution for long-horizon agent harnesses. *arXiv preprint arXiv:2608.24876*, 2026. URL https://arxiv.org/abs/2608.24876.

Alex L. Zhang, Tim Kraska, and Omar Khattab. Recursive language models. *arXiv preprint arXiv:2512.24601*, 2025. URL https://arxiv.org/abs/2512.24601.

Kehao Zhang, Shangtong Gui, Sheng Yang, Wei Chen, and Yang Feng. Learning to remember: End-to-end training of memory agents for long-context reasoning. *arXiv preprint arXiv:2602.18493*, 2026. URL https://arxiv.org/abs/2602.18493.

Yusen Zhang, Ruoxi Sun, Yanfei Chen, Tomas Pfister, Rui Zhang, and Sercan Ö. Arik. Chain of agents: Large language models collaborating on long-context tasks. *arXiv preprint arXiv:2406.02818*, 2024. URL https://arxiv.org/abs/2406.02818.

Jun Zhao, Can Zu, Xu Hao, Yi Lu, Wei He, Yiwen Ding, Tao Gui, Qi Zhang, and Xuanjing Huang. LONGAGENT: Achieving question answering for 128k-token-long documents through multi-agent collaboration. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 16310–16324, 2024. URL https://aclanthology.org/2024.emnlp-main.912/.

Yanjun Zhao, Ruizhong Qiu, Tianxin Wei, Yuanchen Bei, Zhining Liu, Lingjie Chen, Ismini Lourentzou, Hanghang Tong, and Jingrui He. Recontext: Recursive evidence replay as LLM harness for long-context reasoning. *arXiv preprint arXiv:2607.02509*, 2026. URL https://arxiv.org/abs/2607.02509.

Jingsheng Zheng, Xinyuan Fang, Jintian Zhang, Zhengke Gui, Huajun Chen, and Ningyu Zhang. OneDayAgent: Towards a long-horizon harness for autonomous agents. *arXiv preprint arXiv:2608.05013*, 2026. URL https://arxiv.org/abs/2608.05013.

Tianyi Zhuang, Chuqiao Kuang, Xiaoguang Li, Yihua Teng, Jihao Wu, Yasheng Wang, and Lifeng Shang. Docpuzzle: A process-aware benchmark for evaluating realistic long-context reasoning capabilities. *arXiv preprint arXiv:2502.17807*, 2025a. URL https://arxiv.org/abs/2502.17807.

Yufan Zhuang, Xiaodong Yu, Jialian Wu, Ximeng Sun, Ze Wang, Jiang Liu, Yusheng Su, Jingbo Shang, Zicheng Liu, and Emad Barsoum. Self-taught agentic long-context understanding. *arXiv preprint arXiv:2502.15920*, 2025b. URL https://arxiv.org/abs/2502.15920.

Table 4: Benchmark sizes, evidence access, and evaluation metrics. The AA-LCR document count covers all document sets, and LongBench v2 lengths are measured in words from the preprocessed contexts used in our experiments.

Benchmark	# Items	Evidence access	Metric
OfficeQA Pro V1	133	697-doc full corpus	Ex., @5%
OfficeQA Pro V2	90	1,435-doc full corpus	Ex., @5%
AA-LCR-1.1	100	234 provided docs	Judged Ex.
LongBench v2 Long	108	132K–1.38M-word context	Choice Ex.

A Benchmark Details

Table 4 summarizes the four evaluation settings, covering external-corpus retrieval in OfficeQA Pro and reasoning over provided contexts in AA-LCR and LongBench v2.

OfficeQA Pro. OfficeQA Pro V1 (Opsahl-Ong et al., 2026) contains 133 questions over 697 Treasury documents totaling approximately 100M tokens, while V2 contains 90 questions over an expanded corpus of 1,435 documents. V2 places greater emphasis on composite numerical questions requiring values from different series, versions, or documents. Both versions expose the full corpus without question-specific gold documents, requiring the system to identify the sources needed for each answer. We report exact match, which requires the requested value and format, and accuracy within 5% relative error.

AA-LCR-1.1. We use AA-LCR-1.1 (Artificial Analysis Team, 2026),[1] which contains 100 questions associated with 30 document sets comprising 234 documents in total, with an average of 99,325 tokens per set. The documents include company and industry reports, government consultations, academic papers, legal and marketing materials, and survey reports. Questions require synthesis, temporal reasoning, conditional logic, or numerical computation across documents rather than direct retrieval from a single passage. Each question supplies its relevant document set, allowing the evaluation to focus on evidence use within the provided context. For direct-reasoning models, we report the AA-LCR scores published by Artificial Analysis in its official evaluation results.

LongBench v2. LongBench v2 (Bai et al., 2024) contains 503 four-option questions covering single-document QA, multi-document QA, long in-context learning, long-dialogue history understanding, code-repository understanding, and long structured-data understanding. The full benchmark spans contexts of 8K–2M words, most below 128K words. We evaluate the official Long subset of 108 questions, whose contexts in the release used here range from approximately 132K to 1.38M words with a median of 204K words. This subset contains 29 code-repository understanding questions, 26 long in-context learning questions, 24 single-document QA questions, 23 multi-document QA questions, and 6 structured-data understanding questions. Each question provides a long context and four answer options, and performance is measured by exact accuracy on the selected option.

B Method Implementation Details

This appendix specifies the interaction interface, EvidenceBank operations, state contract, and resource settings used in the reported LongHarness evaluations.

B.1 Interaction Interface

The interface distinguishes candidate discovery from source inspection, with `search` returning candidate locations and `read` returning source text together with registered references. Search snippets guide subsequent reading, while statements proposed to the EvidenceBank must cite references registered by `read`.

[1] https://huggingface.co/datasets/ArtificialAnalysis/AA-LCR

Table 5: LONGHARNESS interaction interface, showing the purpose and returned observation of each operation.

Operation	Purpose	Returned observation
`search`	Locate documents or passages relevant to an information need	Ranked candidate locations and navigation snippets
`read`	Inspect selected locations under a bounded read extent	Source text and registered source references
EVIDENCEBANK	Commit or revise source-linked statements	Active statements, revision status, and validation feedback
`compute`	Evaluate a deterministic numerical expression over evidence values	The computed value and integrity warnings
`final`	Submit the answer with supporting references	Accepted output with resolved references, or validation feedback

Table 6: EVIDENCEBANK operations for admitting, correcting, and retracting source-linked statements.

Operation	Input	Effect
`upsert`	One or more proposed source-linked statements	Accepts each validated statement as an active EVIDENCEBANK entry
`revise`	An active evidence identifier and either a replacement statement or a null replacement	Replaces or withdraws the identified active entry

The policy selects one operation per turn, and the controller validates its structure, source references, and resource requirements before execution. Invalid actions return an error observation without modifying the persistent states.

B.2 EVIDENCEBANK OPERATIONS

An `upsert` validates each proposed statement independently by checking that its source references were registered by earlier `read` actions, so a batch may be partially accepted. The controller validates reference registration, while the policy judges whether the statement faithfully represents its sources.

Accepted EVIDENCEBANK entries retain immutable identifiers, and corrections are recorded through `revise` rather than in-place edits. A non-null replacement creates a corrected active entry and marks the previous entry as superseded, while a null replacement retracts the previous entry without creating a replacement. Superseded and retracted entries remain in the revision record but cannot be cited in compute or final requests. Existing conclusions may retain their references until the policy updates them.

B.3 STATE CONTRACT

The CONCLUSION STATE contains established claims and supported hypotheses linked to active EVIDENCEBANK entries, together with unresolved gaps describing information that could change those conclusions. This representation separates the agent's current inferences from the source-linked statements maintained in the EVIDENCEBANK.

When an action updates the EVIDENCEBANK, conclusion revision follows observation of the update result so that claims reference accepted evidence. For other actions, the proposed conclusion update is merged before execution. The complete interaction history remains available alongside the persistent evidence and conclusion states throughout the episode, and final supporting references must resolve to active EVIDENCEBANK entries and their underlying source text. Conclusion updates replace the supplied lists of established claims, hypotheses, or open gaps. Omitted lists remain unchanged. When evidence becomes inactive, transition logs flag affected conclusion references, but the runtime does not automatically delete or rewrite those claims or block subsequent search and

Table 7: Interaction and reading limits used in the LONGHARNESS evaluations.

Setting	Value
Maximum turns	200
Maximum search calls	60
Cumulative read budget	1,000,000 tokens
Maximum tokens per read	16,384

reading. Compute and final validate the evidence IDs declared in their requests, rather than all references retained in the CONCLUSION STATE. Compute also does not establish that literal operands match the cited source values.

B.4 RUNTIME SETTINGS

At each step, the policy receives the system instructions and task specification, followed by the complete interaction history and the current runtime state. The history includes all previous reasoning, actions, and tool observations. Historical observations are retained in full rather than replaced with summaries or placeholders. The runtime state presents the EVIDENCEBANK, CONCLUSION STATE, and remaining resource budgets. These explicit states provide an organized record of acquired evidence and unresolved information needs alongside the full interaction history. The cumulative read budget counts only source text returned by the `read` operation across the episode.

The cumulative reading budget limits source acquisition over the episode; each model request is separately subject to the endpoint's context limit. The client does not automatically truncate or compact the interaction history. A request failure that persists after the model client's retry handling, including rejection of an oversized request, is recorded as `model_request_failed` and terminates the episode. Episodes that terminate without an accepted final answer are scored as incorrect.

The same state and action contract applies across benchmarks, with OfficeQA Pro exposing the full external corpus and AA-LCR and LongBench v2 exposing each question's provided context as the available document collection. Reference answers, gold source annotations, and question-specific document labels are withheld from the agent. All backbone models use temperature 1, top-p 0.95, maximum reasoning effort, and an output limit of 32,768 tokens per request.

B.5 INTERACTION ALGORITHM

Algorithm 1 summarizes the interaction process, with Merge replacing each supplied conclusion-state list and preserving omitted lists (Appendix B.3). For example, an update containing only `open` replaces the unresolved gaps while retaining the existing `established` and `hypotheses` lists; an explicitly empty list clears that field.

Algorithm 1 LONGHARNESS with Evidence–Conclusion Coevolution

Require: Question q, documents D, policy π_θ, budget B_0
Ensure: Accepted final output, or $\perp$ if none is accepted

Initialize document environment $\mathcal{E}_D$, document map M_D, and state x_0
$E_0 \leftarrow \emptyset, C_0 \leftarrow \emptyset, H_0 \leftarrow \emptyset, t \leftarrow 0$
while B_t permits another turn and the environment remains operational **do**
  $S_t \leftarrow (E_t, C_t, H_t)$
  $(a_t, \widehat{C}_t) \sim \pi_\theta(\cdot \mid q, M_D, S_t, B_t)$
    ▷ Select action and optional conclusion update
  $C_{t+1} \leftarrow C_t$
  if a_t violates protocol, source-reference, or budget constraints **then**
    $x_{t+1} \leftarrow x_t$
    $o_{t+1} \leftarrow$ validation feedback
      ▷ Preserve persistent states on invalid actions
  else
    if a_t is neither `upsert` nor `revise` **then**
      $C_{t+1} \leftarrow \mathrm{Merge}(C_t, \widehat{C}_t)$
        ▷ Merge conclusion updates before other actions
    end if
    $(x_{t+1}, o_{t+1}) \sim \mathcal{P}_D(\cdot, \cdot \mid x_t, a_t)$
      ▷ Evidence updates are observed before conclusion revision
  end if
  $E_{t+1} \leftarrow \mathrm{ActiveEvidence}(x_{t+1})$
  Extend H_t with the turn's reasoning, action, and observation to obtain H_{t+1}
  $B_{t+1} \leftarrow$ remaining budget after the turn
  if o_{t+1} confirms an accepted final submission **then**
    return the final output and resolved source references
  end if
  $t \leftarrow t + 1$
end while
return $\perp$

C EXPERIMENTAL PROTOCOL DETAILS

C.1 BENCHMARK PREPARATION

OfficeQA Pro exposes the complete V1 or V2 corpus as an external document environment, with a document map containing identifiers, titles, and section structure. Agents must select relevant sources from this corpus without access to question-specific gold documents. For OfficeQA Pro V1 and V2, we use the Parsed Documents provided by the benchmark authors.

AA-LCR-1.1 retains the document set supplied with each question, including its official grouping and document order. Direct-reasoning baselines receive these documents in the prompt, while agent harnesses access the same collection through search and bounded reading. For LongBench v2, direct-reasoning baselines follow the official truncation procedure with a 256K-token limit, while harnesses access the original Long-subset context through the external document interface.

Reference answers, gold evidence annotations, and question-specific document labels are excluded from agent-facing inputs, and document filtering, ranking, and truncation do not use answer information or the final prediction.

Document structure and retrieval. LONGHARNESS constructs its document environment from the benchmark text, preserving source character offsets for subsequent evidence references. Existing section annotations are retained. Otherwise, sections are identified from Markdown headings, with a single root section used for documents without headings. Searchable blocks follow paragraph boundaries, with oversized paragraphs split into non-overlapping blocks of at most 1,024 estimated tokens, preferably at newline boundaries. Token estimates count word sequences and individual non-whitespace punctuation symbols.

The search interface supports lexical and regular-expression queries, optionally restricted to specified documents or sections. Lexical queries use BM25-style ranking over the text blocks. Search

Table 8: Available operations and resource limits for direct reasoning and the agent harnesses in the main comparison.

System	Available operations	Constraints
Context window	Direct prompt-based reasoning over the supplied or retrieved context	No tool interaction; 256K-token context window
Claude Code	Workspace file search and reading, local shell commands, and workspace file modification	200 model turns and 200 tool calls
Codex	Workspace file inspection, local shell commands, and workspace file modification	200 model turns and 200 tool calls
RLM	Persistent Python REPL, programmatic document access, and model subcalls	30 root-level iterations
LONGHARNESS	`search`, bounded `read`, EVIDENCEBANK `upsert` and `revise`, `compute`, and `final`	200 turns, 60 search calls, a 1M-token cumulative read budget, and at most 16,384 tokens per read

results provide source locators and short snippets, which guide subsequent `read` operations. Read operations return source text together with registered source-span identifiers, allowing EVIDENCEBANK entries to refer back to the corresponding document offsets.

C.2 HARNESS CONFIGURATION

All harnesses operate in an isolated workspace containing the question, available documents, and an output directory. Tool-level network access is disabled, and file access is confined to the workspace, excluding reference answers, scoring scripts, and external run metadata.

We use Codex CLI version 0.135.0 and Claude Code version 2.1.216. Claude Code and Codex inspect and modify workspace files through their native tools and local shell commands. RLM accesses documents programmatically through its Python REPL. LONGHARNESS accesses documents through its defined search and reading operations and maintains evidence and conclusion states through explicit update actions. Table 8 summarizes these interfaces and resource limits. Each comparison uses the same questions, available documents, and backbone model, with the harness interface and execution protocol determining how the model interacts with the corpus. RLM operates in a local Python REPL whose variables persist across iterations within an episode. The model can inspect corpus files, process retrieved text programmatically, and invoke model subcalls through the REPL. The task prompt specifies the question, corpus location, and required answer format. Our benchmark adapter converts document paths to absolute paths accessible from the REPL and adds the benchmark-specific output contract. The root loop allows up to 30 iterations. The driver uses the library's default maximum recursion depth of one, at which subcalls receive plain model completions rather than launching further REPL loops. Root and subcalls use the same backbone. If the returned answer does not satisfy the required output format, the adapter makes one additional call to the same model, instructed to copy the answer value verbatim into the required wrapper.

Evidence-pipeline audit. We use GPT-5.6-Luna with temperature 0 to assess incorrect episodes using the recorded reasoning and available tool outputs or evidence-state records. The audit distinguishes failures to retrieve the required evidence, failures to select or retain that evidence, and failures to transform the adopted evidence into a correct answer. Answer correctness is determined by the benchmark scorer at tolerance 0; correct episodes are counted as passing the first two stages by protocol. Retrieval coverage is computed over all evaluated questions, evidence adoption over questions passing retrieval, and answer conversion over questions passing both preceding stages. These conditional rates characterize stage outcomes within the analyzed runs.

C.3 LONGHARNESS PROMPT TEMPLATE

The template below presents the instructions used by LONGHARNESS in the terminology of the paper. A task-specific block supplies the question, answer requirements, document map, tool schemas, and resource budget, with the available operations defined in Appendix B.1.

Runtime request construction. The prompt template supplies the policy instructions. Each model request additionally contains a task block specifying the question, answer requirements, document map, and action schemas. This is followed by all previous assistant responses and tool observations in chronological order, then a current-state block containing the EVIDENCEBANK, CONCLUSION STATE, and remaining budgets. The CONCLUSION STATE is serialized under the `state` field, with `established`, `hypotheses`, and `open` lists. Historical observations remain available alongside these explicit states.

LongHarness Prompt Template

```
You are a ReAct agent for evidence-grounded long-context question
answering. Documents remain external to the prompt. Investigate them over
multiple turns using search, read, compute, EvidenceBank, and final. You
decide what the evidence means, what remains uncertain, and when the answer
is sufficient.

Follow one revisable loop:

1. Select the unresolved gap whose resolution is most likely to change the
   answer.
2. Use search to locate relevant documents and passages. Search snippets are
   navigation aids, not citable evidence.
3. Use read to inspect source text and register source references. Only a
   source reference returned by read may support an EvidenceBank statement.
4. Save useful source facts as atomic, source-linked EvidenceBank statements.
   An EvidenceBank identifier becomes usable only after a successful
   EvidenceBank observation.
5. Only cite EvidenceBank identifiers that were active before the current
   turn. Evidence created by the current EvidenceBank action becomes citable
   on the next turn.
6. Revise the Conclusion State when new evidence supports, qualifies,
   contradicts, or replaces earlier reasoning.

The EvidenceBank records what the documents state, not whether an answer is
correct. Keep statements atomic and source-faithful. Retry only rejected
statements. If an accepted statement is invalid, use EvidenceBank revise:
provide a corrected replacement, or use a null replacement to withdraw it.

For compute, place the source values in one deterministic expression and cite
the active EvidenceBank identifiers that support those values. After compute,
verify direction, unit, and precision. Express the derived result in the
Conclusion State together with the EvidenceBank identifiers supporting every
operand.

Maintain a compact Conclusion State with three revisable lists:

- established: conclusions justified by active EvidenceBank entries;
- supported hypotheses: plausible but unsettled interpretations supported by
  active EvidenceBank entries;
- unresolved: concrete missing information that could change the answer.

Established conclusions and supported hypotheses must cite active EvidenceBank
entries. Unsupported possibilities belong under unresolved. State the
inference, not merely disconnected source facts.

Search for the missing distinction, not only for the question topic. If
retrieval is uninformative, change the query or reading strategy instead of
repeating it. Read enough context to interpret the source correctly. Before
adopting a conclusion, verify that its entities, relationships, scope, and
requested output match the question. A derived or aggregate conclusion must
cite every decision-critical premise and use compute when arithmetic is
needed. Evidence relevance alone is not support.

Before final, reconstruct the requested answer only from current established
conclusions and their sources. Revise stale conclusions. Check a plausible
alternative only when it could change the answer. Continue only while
resolving a remaining unresolved gap could change the result. Do not search
merely because unrelated uncertainty remains.

Place the answer-bearing conclusion under established. Submit final with its
active EvidenceBank identifiers. Follow the requested choice label, units,
precision, ordering, and output format.

Invoke exactly one operation per turn. Conclusion State updates may be
partial, and omitted lists retain their previous values. Evidence mutation
turns defer Conclusion State updates until their observations have been seen.
Use final as the only way to submit the answer.
```

Table 9: Three-run Exact accuracy with GLM-5.3-Flash. Run scores, means, and maxima are percentages. Standard deviations are in percentage points.

Harness	Run 1	Run 2	Run 3	Mean	Std.	Max.
OfficeQA Pro V1						
Codex	45.86	46.62	47.37	46.62	0.76	47.37
Claude Code	48.12	48.87	51.13	49.37	1.57	51.13
RLM	45.11	42.86	48.12	45.36	2.64	48.12
LongHarness	60.15	60.15	59.40	59.90	0.43	60.15
OfficeQA Pro V2						
Codex	36.67	42.22	44.44	41.11	4.00	44.44
Claude Code	27.78	34.44	35.56	32.59	4.21	35.56
RLM	38.89	31.11	38.89	36.30	4.49	38.89
LongHarness	61.11	58.89	58.89	59.63	1.28	61.11
AA-LCR 1.1						
Codex	72.00	77.00	80.00	76.33	4.04	80.00
Claude Code	79.00	79.00	83.00	80.33	2.31	83.00
RLM	52.00	58.00	56.00	55.33	3.06	58.00
LongHarness	86.00	84.00	82.00	84.00	2.00	86.00
LongBench v2 Long						
Codex	68.52	69.44	69.44	69.13	0.53	69.44
Claude Code	62.04	62.96	64.81	63.27	1.41	64.81
RLM	59.26	60.19	62.96	60.80	1.92	62.96
LongHarness	76.85	75.93	73.15	75.31	1.93	76.85

D Repeated-Run Results

Table 9 reports three runs for each harness and evaluation setting with GLM-5.3-Flash. Summary statistics are computed from the three reported scores. Std. denotes the sample standard deviation, and Max. is the highest score across the three runs.

E Case Study

Figure 4 compares GLM-5.3-Flash trajectories for computing the geometric mean of 80 monthly Treasury expenditures in OfficeQA Pro V1. We randomly selected this case from Codex trajectories that failed the exact-match criterion. Steps in Figure 4 denote shell-command calls for Codex and policy actions for LongHarness. Codex locates the relevant table within 22 command calls but leaves missing values implicit, later making 103 consecutive backward reads (*unbounded exploration*). It computes 4958.09 using 4100 for May 1942, then recognizes the corrected value 4400 but retains the earlier result in its final answer. Replacing this operand yields the reference answer 4962.46. Codex therefore fails exact match but passes the 5% tolerance criterion, with a relative error of approximately 0.088%. This *evidence-state drift* illustrates a recognized correction that was not incorporated into the final computation.

After recording 4400 in the EvidenceBank, LongHarness tracks cross-issue consistency as an unresolved Conclusion State gap. Further reading corroborates the value and corrects two other entries. Evidence and conclusions incorporate these revisions before computation. LongHarness returns 4962.46 in 10 policy actions, showing how conclusion-guided retrieval and evidence maintenance carry corrections into the final decision.