SRCV: An Enhanced Harness for Evidence-Grounded Agentic Reasoning

Zixuan Wang
Department of Computer Science
Johns Hopkins University

1 Introduction

Large language model (LLM) agents can solve complex tasks by interacting with external tools to acquire evidence and perform multi-step reasoning. In legal reasoning, DeonticBench (Dou et al., 2026a) evaluates whether models can apply explicit rules to case-specific facts, while DAR (Dou et al., 2026b) extends this setting with agentic harnesses for accessing statutes and executing computations. The agent must examine a given statute, identify the applicable provisions, conditions, and exceptions, and reason about their implications for the case. In contrast, BrowseComp-Plus (Chen et al., 2025) evaluates agents on complex questions that require searching a large collection of web documents and combining evidence from multiple sources. While the former emphasizes selecting and applying rules from a given document, the latter emphasizes discovering relevant evidence across a much larger search space.

Existing agent harnesses, such as Terminus-2 (Merrill et al., 2026) and KIRA (KRAFTON AI & Ludo Robotics, 2026), largely rely on a single agent to retrieve, reason, and answer. As the agent interacts with tools, its context accumulates search queries, retrieved documents, intermediate reasoning, and repeated tool outputs, making it difficult to maintain relevant evidence throughout long trajectories. Recent work, such as SelfCompact (Li et al., 2026a) and Context Language Models (Shao et al., 2026), explores adaptive context management to address growing interaction histories. However, these approaches primarily manage information after it has entered the agent's context, rather than controlling which information enters the main reasoning trajectory in the first place, which allows irrelevant evidence and redundant tool outputs to accumulate before compaction is triggered, potentially compromising the agent's ability to manage its context effectively.

To address this limitation, we develop a **Solver–Reader–Compaction–Verifier (SRCV) Harness** that combines proactive and reactive context management. The reader proactively selects relevant evidence while keeping its intermediate retrieval process outside the solver's main trajectory. The compaction module subsequently manages the accumulated context by preserving essential evidence and reasoning while removing redundant information. A rubrics-based verifier examines candidate answers against the source evidence and provides feedback for revision. The design is implemented at the harness level, allowing it to be integrated with different base agents.

We evaluate SRCV on DeonticBench through DAR and on BrowseComp-Plus, using Terminus-2, KIRA, and Mini-SWE-Agent as base harness. Preliminary results show that SRCV improves SARA-Numeric accuracy from 22.9% to 48.6% on Terminus-2 and from 11.4% to 51.4% on KIRA. On BrowseComp-Plus SRCV improves accuracy from 28.0% to 52.0% compared with CLM-Agent. However, these improvements come with additional inference cost. Our next step is to investigate the accuracy–cost trade-off and improve the efficiency.

2 Method

2.1 Architecture Overview

We introduce SRCV, a harness that combines proactive evidence selection, reactive context compaction, and reasoning verification. Given a task x and an external evidence source $\mathcal{D}$, SRCV organizes the agent's workflow into four roles: Solver, Reader, Compaction, and Verifier. All four roles use the same underlying language model, with role-specific instructions and skills. The solver maintains the main reasoning trajectory, while the reader and compaction module operate on branches

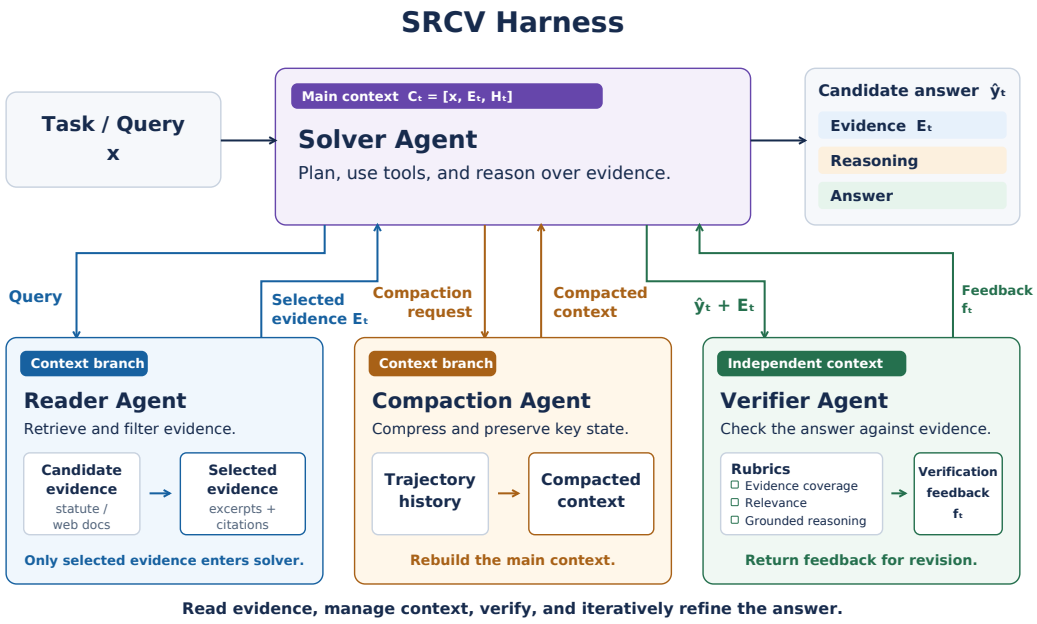

Figure 1: Overview of SRCV (Solver–Reader–Compaction–Verifier) Harness.

of its context. The verifier independently examines candidate answers against the source evidence. This separation allows harness to control the information entering the main agent's context, manage its accumulated history, and provide feedback for answer revision.

2.2 Proactive and Reactive Context Management

Solver. The solver S_θ is the main agent responsible for planning, tool calling, and reasoning to produce the final answer. At step t, it maintains a working context $\mathcal{C}_t = [x, \mathcal{E}_t, \mathcal{H}_t]$, where x is the task, $\mathcal{E}_t$ denotes already retrieved evidence, and $\mathcal{H}_t$ contains intermediate reasoning result. The solver can request additional evidence from the reader through a query q_t, or invoke compaction when its context becomes redundant or approaches the context limit.

Reader: Proactive Context Control.

Rather than allowing the main agent to directly accumulate all retrieved documents, SRCV delegates evidence acquisition to a reader agent R_θ. Given a query q_t from the solver, the reader searches the external source $\mathcal{D}$ and retrieves evidence relevant to the current task. We formalize this process as

$$\begin{aligned} \mathcal{E}_t &= R_\theta(x, q_t, \mathcal{D}; \mathcal{C}_t), \\ \mathcal{C}_{t+1} &= \mathcal{C}_t \oplus \mathcal{E}_t \end{aligned} \tag{1}$$

where $\mathcal{E}_t$ denotes the retrieved evidence and $\oplus$ denotes adding new evidence to the solver's context. In DeonticBench, the reader identifies applicable statutes, including relevant conditions and exceptions. In BrowseComp-Plus, it searches a large document collection for evidence relevant to the query. The harness delivers the selected evidence to the solver, provides proactive control over context growth by filtering information before it enters the solver's context.

In BrowseComp-Plus, we implement a selective reader as in Figure 2 that initially examines only the first 512 tokens of each retrieved document and assesses its relevance to the task. When further inspection is needed, the reader can selectively retrieve additional line ranges (`OPEN`) or search for specific phrases (`FIND`), with at most three additional actions per document. Only the selected source passages are forwarded verbatim to the solver, while the reader's intermediate processing remains outside the solver's context. Further details on the reader instructions and evidence delivery protocol are provided in Appendix B.1.

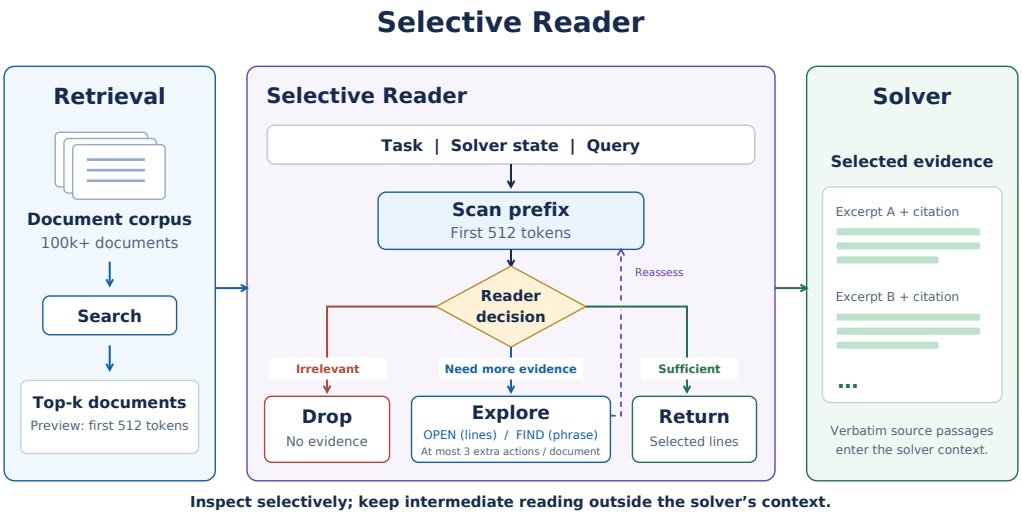

Figure 2: Overview of the selective reader. The reader inspects document prefixes and selectively explores relevant documents before passing selected evidence to the solver.

Compaction: Reactive Context Management. Although the reader limits full documents entering the main trajectory, the solver's context continues to grow through multi-turn retrieved evidence and reasoning steps. SRCV therefore introduces a complementary compaction module M_θ that reorganizes the accumulated context. When invoked by the solver, the compaction agent identifies the task state, essential evidence, and intermediate results needed for subsequent process. The harness then reconstructs the working context as

$$\mathcal{C}'_t = [x,\ M_\theta(\mathcal{C}_t),\ M_\theta(\mathcal{E}_t)] \tag{2}$$

where M_θ denotes compaction agent summarizing the current reasoning state while $M_\theta(\mathcal{E}_t)$ selects essential evidence from previously retrieved information.

Together, the reader and compaction agents provide proactive and reactive context management: controls which information enters the trajectory, while manages information already accumulated.

Prefix Caching. To reduce the inference overhead introduced by subagents, we leverage prefix KV-cache reuse. On DAR, the reader and compaction agents operate as branches of the solver's context, sharing its existing prefix rather than recomputing the entire trajectory. On BrowseComp-Plus, successive selective reader actions reuse the cached prefix of the same reading session. This design reduces redundant prefill computation and mitigates the additional inference cost of SRCV.

2.3 Self-Verification

Verifier: Rubrics-Grounded Verification. Before finalizing an answer, SRCV invokes a verifier V_θ to examine the solver's candidate answer against the source evidence. Using three complementary rubrics, the verifier assesses evidence coverage, alignment between the evidence and task, and the logical consistency of the reasoning and final answer. The verification and revision process is expressed as

$$\begin{aligned} f_t &= V_\theta(x, \hat{y}_t, \mathcal{E}_t, \mathcal{D}), \\ \hat{y}_{t+1} &= S_\theta(\mathcal{C}_t \oplus f_t) \end{aligned} \tag{3}$$

where $\hat{y}_t$ is the candidate answer and f_t is the verification feedback. The feedback identifies potentially missing evidence or unsupported reasoning and allows the solver to revise its answer.

3 EXPERIMENTS

3.1 SETTING

Models. We evaluate SRCV using two open-weight language models: Qwen3.5-35B-A3B (Qwen Team, 2026) for DAR and Qwen3.5-9B for BrowseComp-Plus. All models are served locally using vLLM 0.20.1 on an NVIDIA DGX Spark. For DAR, we use BF16 precision, a 65k-token model context window, and a maximum of 16k generated tokens per turn, with a 32k-token compaction threshold. We enable thinking mode and use temperature 0.7, top-p 0.95, and top-k 20. For BrowseComp-Plus, we follow the CLM-Agent setup with a 28k-token context budget and a maximum of 100 agent steps.

Base Harnesses. On DAR, we evaluate SRCV with two base harnesses: Terminus-2 (Merrill et al., 2026) and Terminus-KIRA (KRAFTON AI & Ludo Robotics, 2026). Terminus-2 uses shell-based tool execution, while KIRA extends it with native tool calling. We compare SRCV against their original implementations under the DAR evaluation framework (Harbor Framework Team, 2026). On BrowseComp-Plus, we build upon the released CLM-Agent (Shao et al., 2026), which uses a Mini-SWE-Agent-based execution loop (Yang et al., 2024) and allows the model to edit its accumulated context. SRCV extends these base harnesses with dedicated reader and verifier roles, while retaining their underlying agent execution mechanisms.

Benchmarks. We evaluate SRCV on two complementary benchmarks. For deontic reasoning, we use the SARA hard split of DeonticBench (Dou et al., 2026a) SARA-Numeric contains 35 cases requiring numerical tax calculations, while SARA-Binary contains 30 cases requiring *Entailment* or *Contradiction* judgments. For large-scale information retrieval, we use BrowseComp-Plus, which consists of 830 challenging questions over a fixed corpus of approximately 100k web documents. Our preliminary evaluation uses a fixed 50-question subset. We use Qwen3-Embedding-8B with the released retrieval index to retrieve the top 10 documents per search.

Evaluation Protocol. We report exact-match accuracy for SARA-Numeric and macro-F1 for SARA-Binary. For BrowseComp-Plus, we evaluate answer correctness using Qwen3.5-35B-A3B as a judge with temperature 0, following the benchmark's grading template. Missing answers are counted as incorrect. Alongside task performance, we measure inference cost to characterize the accuracy–cost trade-off. For DAR, we estimate cost from token usage using a fixed pricing scheme. For BrowseComp-Plus, we report average PFLOPs per question, accounting for prefix KV-cache reuse by counting shared prompt prefixes only once per trial.

3.2 RESULT

Table 1: Performance and estimated cost on SARA with Qwen3.5-35B-A3B. Scores are percentages; numeric accuracy includes correct-case counts. Costs are US cents per case with prompt caching. Bold indicates the better score within each harness.

Harness	**Method**	**SARA-Numeric**		**SARA-Binary**	
		Accuracy ↑	Cost ↓	Macro-F1 ↑	Cost ↓
Terminus-2	Baseline	22.9 (8/35)	1.54	**83.3**	0.59
	SRCV (ours)	**48.6** (17/35)	4.74	79.9	3.04
Terminus-KIRA	Baseline	11.4 (4/35)	1.37	82.9	0.78
	SRCV (ours)	**51.4** (18/35)	4.40	**86.6**	2.77

Table 2: Performance and compute cost on BrowseComp-Plus with Qwen3.5-9B in the zero-shot setting. The baseline is our run of the released CLM-Agent. Compute is reported as average PFLOPs per question.

Harness	**Method**	**Accuracy (%) ↑**	**PFLOPs/question ↓**
Mini-SWE-Agent	CLM-Agent	28.0 (14/50)	**3.20**
	SRCV (ours)	**52.0** (26/50)	3.90

Table 3: Failure analysis on the BrowseComp-Plus subset. Termination reasons among unsuccessful trials.

Failure Type	CLM-Agent	SRCV (ours)
Fail to compact	32	**0**
Step limit	4	24
Incorrect submission	0	0
Total failures	36	**24**

Table 4: Evidence acquisition efficiency on BrowseComp-Plus.

Metric	CLM-Agent	SRCV (ours)	Change
Documents examined / task ↑	61.8	**140.9**	$2.28\times$
Solver context (chars / result) ↓	1,936	**236**	-87.8%

Table 1 shows that SRCV improves SARA-Numeric accuracy by 25.7 and 40.0 percentage points on Terminus-2 and Terminus-KIRA, respectively. These substantial gains across two distinct base harnesses demonstrate the effectiveness of SRCV in identifying and applying relevant statutory provisions through structured evidence retrieval and verification. The consistent improvements further highlight the adaptability of SRCV across different agent harness.

On BrowseComp-Plus (Table 2), SRCV improves accuracy from 28.0% to 52.0% compared with CLM-Agent. Trajectory analysis suggests two main benefits of reader-mediated evidence acquisition. First, filtering retrieved documents before they enter the solver's context reduces context accumulation. As shown in Table 3, 32 of the 36 unsuccessful CLM-Agent trials terminated after exhausting context-limit recovery attempts, compared with none under SRCV. This suggests that proactive evidence selection reduces reliance on model-initiated compaction, which can become unreliable as the context grows. Second, reader-mediated retrieval enables broader document exploration under a fixed context budget. As shown in Table 4, SRCV examines 2.28 times as many documents per question while reducing the solver's context load per search result by 87.8%. These findings highlight the effectiveness of combining proactive evidence selection with reactive context compaction.

However, these improvements come with additional inference cost, motivating further improvement of cost efficiency.

4 Challenges and Next Steps

Inference Cost. Although SRCV improves answer accuracy, its additional agent calls introduce substantial computational overhead. We plan to investigate more efficient KV-cache reuse and adaptive stopping strategies, inspired by agentic abstention (Luo et al., 2026), to reduce unnecessary tool calls and improve the accuracy–cost trade-off.

Confidence Estimation. We further aim to estimate answer reliability from the verifier's assessments. However, preliminary experiments show that both self-verbalized confidence and verifier rubric scores poorly distinguish correct from incorrect answers (Appendix C). We plan to explore representation-based confidence estimation using the verifier's hidden states.

References

Zijian Chen, Xueguang Ma, Shengyao Zhuang, Ping Nie, Kai Zou, Andrew Liu, Joshua Green, Kshama Patel, Ruoxi Meng, Mingyi Su, Sahel Sharifymoghaddam, Yanxi Li, Haoran Hong, Xinyu Shi, Xuye Liu, Nandan Thakur, Crystina Zhang, Luyu Gao, Wenhu Chen, and Jimmy Lin. Browsecomp-plus: A more fair and transparent evaluation benchmark of deep-research agent, 2025. URL https://arxiv.org/abs/2508.06600.

Guangyao Dou, Luis Brena, Akhil Deo, William Jurayj, Jingyu Zhang, Nils Holzenberger, and Benjamin Van Durme. DeonticBench: A benchmark for reasoning over rules. *arXiv preprint*

arXiv:2604.04443, 2026a. doi: 10.48550/arXiv.2604.04443. URL https://arxiv.org/abs/2604.04443.

Guangyao Dou, William Jurayj, Nils Holzenberger, and Benjamin Van Durme. DAR: Deontic reasoning with agentic harnesses. *arXiv preprint arXiv:2606.05009*, 2026b. doi: 10.48550/arXiv.2606.05009. URL https://arxiv.org/abs/2606.05009.

Shihan Dou, Ming Zhang, Zhangyue Yin, Chenhao Huang, Yujiong Shen, Junzhe Wang, Jiayi Chen, Yuchen Ni, Junjie Ye, Cheng Zhang, Huaibing Xie, Jianglu Hu, Shaolei Wang, Weichao Wang, Yanling Xiao, Yiting Liu, Zenan Xu, Zhen Guo, Pluto Zhou, Tao Gui, Zuxuan Wu, Xipeng Qiu, Qi Zhang, Xuanjing Huang, Yu-Gang Jiang, Di Wang, and Shunyu Yao. Cl-bench: A benchmark for context learning, 2026c. URL https://arxiv.org/abs/2602.03587.

Harbor Framework Team. Harbor: A framework for evaluating and optimizing agents and models in container environments, 2026. URL https://doi.org/10.5281/zenodo.20953922.

KRAFTON AI and Ludo Robotics. Terminus-kira: Boosting frontier model performance on terminal-bench with minimal harness, 2026. URL https://github.com/krafton-ai/kira.

Tianjian Li, Jingyu Zhang, William Jurayj, Xi Wang, Chuanyang Jin, Mehrdad Farajtabar, Eric Nalisnick, and Daniel Khashabi. Self-compacting language model agents, 2026a. URL https://arxiv.org/abs/2606.23525.

Xiaochuan Li, Ryan Ming, Meng Chu, Shuai Shao, Rong Jin, and Chenyan Xiong. Acm: Agentic context management for long horizon tasks, 2026b. URL https://arxiv.org/abs/2607.23809.

Han Luo, Bingbing Wen, and Lucy Lu Wang. Agentic abstention: Do agents know when to stop instead of act?, 2026. URL https://arxiv.org/abs/2606.28733.

Mike A Merrill, Alexander Glenn Shaw, Nicholas Carlini, Boxuan Li, Harsh Raj, Ivan Bercovich, Lin Shi, Jeong Yeon Shin, Thomas Walshe, E. Kelly Buchanan, Junhong Shen, Guanghao Ye, Haowei Lin, Jason Poulos, Maoyu Wang, Jenia Jitsev, Marianna Nezhurina, Di Lu, Orfeas Menis Mastromichalakis, Zhiwei Xu, Zizhao Chen, Yue Liu, Robert Zhang, Leon Liangyu Chen, Anurag Kashyap, Jan-Lucas Uslu, Jeffrey Li, Jianbo Wu, Minghao Yan, Song Bian, Vedang Sharma, Ke Sun, Steven Dillmann, Akshay Anand, Andrew Lanpouthakoun, Bardia Koopah, Changran Hu, Etash Kumar Guha, Gabriel H. S. Dreiman, Jiacheng Zhu, Karl Krauth, Li Zhong, Niklas Muennighoff, Robert Kwesi Amanfu, Shangyin Tan, Shreyas Pimpalgaonkar, Tushar Aggarwal, Xiangning Lin, Xin Lan, Xuandong Zhao, Yiqing Liang, Yuanli Wang, Zilong Wang, Changzhi Zhou, David Heineman, Hange Liu, Harsh Trivedi, John Yang, Junhong Lin, Manish Shetty, Michael Yang, Nabil Omi, Negin Raoof, Shanda Li, Terry Yue Zhuo, Wuwei Lin, Yiwei Dai, Yuxin Wang, Wenhao Chai, Shang Zhou, Dariush Wahdany, Ziyu She, Jiaming Hu, Zhikang Dong, Yuxuan Zhu, Sasha Cui, Ahson Saiyed, Arinbjörn Kolbeinsson, Christopher Michael Rytting, Ryan Marten, Yixin Wang, Alex Dimakis, Andy Konwinski, and Ludwig Schmidt. Terminal-bench: Benchmarking agents on hard, realistic tasks in command line interfaces. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=a7Qa4CcHak.

Vitaliy Polshkov, Marcin Pitera, Jeremy Yang, Kirill Priemko, Maksim Gaiduk, Aleksandr Nikolenko, Denis Bykov, Clare Southern, Denis Yarats, and Jerry Ma. Wandr: A benchmark for wide and deep research, 2026. URL https://arxiv.org/abs/2608.14747.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

Rulin Shao, Shannon Zejiang Shen, Junjie Oscar Yin, Yuetai Li, Minheng Wang, Hamish Ivison, Radha Poovendran, Nathan Lambert, Teng Xiao, Mike Lewis, Wen tau Yih, Luke Zettlemoyer, and Pang Wei Koh. Context language models, 2026. URL https://arxiv.org/abs/2609.37725.

Weiwei Sun, Miao Lu, Zhan Ling, Kang Liu, Xuesong Yao, Yiming Yang, and Jiecao Chen. Scaling long-horizon llm agent via context-folding, 2025. URL https://arxiv.org/abs/2510.11967.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Musique: Multi-hop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022. URL https://aclanthology.org/2022.tacl-1.31/.

Jason Wei, Zhiqing Sun, Spencer Papay, Scott McKinney, Jeffrey Han, Isa Fulford, Hyung Won Chung, Alex Tachard Passos, William Fedus, and Amelia Glaese. Browsecomp: A simple yet challenging benchmark for browsing agents, 2025. URL https://arxiv.org/abs/2504.12516.

Xixi Wu, Kuan Li, Yida Zhao, Liwen Zhang, Litu Ou, Huifeng Yin, Zhongwang Zhang, Yong Jiang, Pengjun Xie, Fei Huang, Minhao Cheng, Shuai Wang, Hong Cheng, and Jingren Zhou. Resum: Unlocking long-horizon search intelligence via context summarization, 2025. URL https://arxiv.org/abs/2509.13313.

John Yang, Carlos E. Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. Swe-agent: Agent-computer interfaces enable automated software engineering, 2024. URL https://arxiv.org/abs/2405.15793.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of EMNLP*, 2018. URL https://aclanthology.org/D18-1259/.

Andrew Zhu, Alyssa Hwang, Liam Dugan, and Chris Callison-Burch. Fanoutqa: A multi-hop, multi-document question answering benchmark for large language models. In *Proceedings of ACL*, pp. 18–37, 2024. URL https://aclanthology.org/2024.acl-short.2/.

A Related Work

A.1 Benchmarks for Evidence-Grounded Reasoning

Rule-Based and Context-Grounded Reasoning. DeonticBench (Dou et al., 2026a) evaluates the application of explicit rules and policies to case-specific facts, while DAR (Dou et al., 2026b) extends this setting to agentic reasoning with on-demand access to statutory documents. More broadly, CL-bench (Dou et al., 2026c) evaluates whether models can learn and apply new knowledge, rule systems, and procedures provided in task-specific contexts. These benchmarks emphasize identifying relevant information and applying it correctly, rather than discovering information through open-ended search.

Multi-Hop Question Answering. Classical benchmarks such as HotpotQA (Yang et al., 2018) and MuSiQue (Trivedi et al., 2022) evaluate reasoning over evidence distributed across multiple documents. HotpotQA provides supporting-fact annotations, while MuSiQue emphasizes compositional reasoning across dependent questions. FanOutQA (Zhu et al., 2024) further evaluates questions that require gathering information about multiple entities across documents. These benchmarks provide established settings for studying evidence retrieval and aggregation.

Long-Horizon Search and Deep Research. BrowseComp (Wei et al., 2025) evaluates agents on difficult questions requiring persistent web search. BrowseComp-Plus (Chen et al., 2025) extends this setting with a fixed document corpus, enabling more reproducible evaluation of retrieval and reasoning. WANDR (Polshkov et al., 2026) moves beyond single-answer questions to wide research tasks requiring the discovery and verification of many records. Together, these benchmarks highlight different challenges in evidence acquisition, from targeted multi-hop retrieval to long-horizon information gathering.

A.2 Agent Context Management

Trajectory Summarization and Editing. ReSum (Wu et al., 2025) addresses context limitations through periodic summarization of accumulated search trajectories. SelfCompact (Li et al., 2026a) allows agents to decide when to summarize their history, guided by lightweight compaction rubrics. Context Language Models (Shao et al., 2026) instead allow agents to directly edit their stored context. These approaches primarily manage information already accumulated during agent execution.

Memory-Augmented Context Management. ACM (Li et al., 2026b) introduces agent-controlled context compression with external memory storage, allowing previously removed information to be retrieved when needed. This reduces working-context pressure while preserving access to earlier interactions.

Sub-Trajectory Isolation. Context-Folding (Sun et al., 2025) enables agents to branch into temporary sub-trajectories for localized tasks and return concise results without retaining their intermediate execution histories in the main context. SRCV adopts a related principle of isolating auxiliary processing, specializing it for evidence acquisition through a reader that returns selected source passages. It further combines this proactive context control with subsequent compaction and independent evidence-grounded verification.

B SRCV Harness Details

B.1 Reader

On BrowseComp-Plus, SRCV employs a selective reader for evidence acquisition. Rather than reading every retrieved document in full, the reader initially inspects a short document prefix and selectively retrieves additional passages when needed. The reader receives the task, the solver's current state, its reading request, and line-numbered source text. This design reduces unnecessary document processing while keeping the reader's intermediate trajectory outside the solver's main context.

Reader Instructions

Task. Identify evidence relevant to the question from a retrieved document, without answering the question.

1. Examine. Initially inspect the first 512 tokens of the document and assess its relevance to the question and the solver's unresolved information needs.

2. Explore. If additional evidence is needed, retrieve specific line ranges using `OPEN` or locate relevant phrases using `FIND`. Perform at most three additional actions per document. Stop when sufficient evidence has been collected or further inspection is unlikely to be useful.

3. Select. Prioritize concrete facts such as names, dates, relationships, and numerical values. Select informative source passages that support or rule out candidate answers. Only select passages that have been inspected.

4. Return. At each turn, assign a relevance grade and return selected source line ranges with brief factual descriptions. Previously selected evidence is retained across turns. Avoid summarizing the entire document.

Output format.

`SCAN`	Brief assessment of document relevance.
`MATCH`	`strong`, `weak`, or `none`.
`LINES`	Selected line ranges with factual labels; at most four ranges and eight lines.
`OPEN`	Optional action to retrieve a specified line range.
`FIND`	Optional action to locate specific phrases within the document.

Evidence delivery. The harness copies selected source passages verbatim into the solver's context, while excluding the reader's intermediate processing.

B.2 COMPACTION

On DAR, the compaction module reorganizes the solver's accumulated trajectory into a concise state representation. Rather than summarizing all information uniformly, it separates the reasoning state from source evidence that must be preserved verbatim.

Compaction State Page

Task. Preserve the information needed to continue the current task while removing redundant interaction history.

State representation.

Established	Verified intermediate results, applicable rules, and supporting computations.
Rejected	Previously considered alternatives and the reasons they were ruled out.
Open	Unresolved questions, missing evidence, and outstanding verification findings.
Next	The next reasoning or retrieval step.

Evidence preservation.
The compaction agent additionally returns `keep_source`, a list of source line ranges needed for subsequent reasoning. The harness retrieves these passages from the original document rather than generating paraphrased versions.

Context Reconstruction. The following schematic illustrates how the accumulated trajectory is reorganized.

Context Before and After Compaction

Before Compaction	After Compaction
Original task	Original task
Retrieved evidence	Structured state page
Intermediate reasoning	Selected evidence (verbatim)
Repeated tool outputs	Current answer and feedback
Earlier attempts and revisions	Recent interactions

The harness reconstructs the solver's context using the generated state page and the selected original evidence. This preserves critical information while removing redundant history from the main trajectory.

B.3 VERIFIER

On DAR, SRCV uses three independent rubrics to verify the solver's candidate answer. Each rubric examines a different aspect of evidence-grounded reasoning and produces a structured assessment.

Evidence-Grounded Verification Rubrics

Rubric	Verification Criteria
Section Table	Identify statutory provisions engaged by the case facts and check whether the answer correctly includes their conditions and exceptions.
Fact Table	Map case facts to the provisions that use them. Identify relevant facts that are omitted, misinterpreted, or applied incorrectly.
Grounded Reasoning	Check whether reasoning claims match the source provisions, recompute numerical results, and verify that the final answer follows from the stated reasoning.

Independent Assessment. Each rubric constructs its assessment from the original evidence and the candidate answer, rather than relying on the solver's intermediate reasoning history.

Structured Verification Report

Rubric Table	Criterion-level assessments supported by source evidence.
Verdict	`PASS` or `FAIL`.
Findings	Identified errors, exact source quotations, and supporting case facts or computations.
Notes	Potential concerns that could not be verified against the available evidence.

Grounding Requirement. A finding must be supported by a verified source quotation or computation. The verifier reports errors without directly supplying the corrected final answer.

On BrowseComp-Plus, the verification criteria are adapted to multi-document reasoning. The verifier checks each condition of the question against cited evidence using `SUPPORTED`, `NOT SUPPORTED`, or `CONTRADICTED`, followed by a consistency check of the final answer.

C Confidence Calibration

At the end of each trajectory, we asked Qwen3.5-35B-A3B to estimate the probability that its answer was correct. On SARA-Numeric, mean confidence ranged from 86% to 90% across harnesses, despite accuracies of 17% to 31% (Table 5). Even at a 90% confidence threshold, only 2 of the 18 retained Terminus-2 answers were correct.

Table 5: Self-verbalized confidence on SARA. ECE uses 10 equal-width bins. Kuiper is confidence-weighted.

Task	Harness	Acc.	Mean conf.	ECE ↓	Kuiper ↓	AUROC ↑
Numeric	Terminus-2	0.171	0.863	0.691	0.610	0.351
	KIRA	0.314	0.900	0.586	0.528	0.538
Binary	Terminus-2	0.867	0.935	0.085	0.075	0.688
	KIRA	0.867	0.945	0.128	0.113	0.375

The verifier's final rubric score also separated correct from incorrect answers poorly, with AUROCs of 0.50 on completed numeric trials and 0.58 on binary trials. Of the 20 numeric answers that received the maximum score, ten were wrong. We will next test whether a linear probe on the verifier's hidden states can better identify cases for expert review.