AUDITING AND REDESIGNING DEEP-RESEARCH REPORT-WRITING PIPELINES FOR FAITHFUL REPORTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep-research (DR) systems are commonly evaluated with report-level rubrics, which can obscure whether their writers omit retrieved evidence, introduce unsupported claims, or attach citations to the wrong sources. We introduce CLAIMPROBE, a claim-level audit that aligns report claims with the evidence collected by a DR system. Applied to three open DR systems, CLAIMPROBE reveals substantial differences in faithfulness and evidence coverage among reports with similar rubric-based scores. These findings motivate CLAIMWRITER, an evidence-linked writer that organizes atomic source facts before drafting report sections and preserves their links to the resulting prose. Across the evaluated host and judge configurations, replacing each system's native writer with a CLAIMWRITER variant lowers measured hallucination by 7 to 21 percentage points and raises necessary-fact recall by 7 to 30 points, while changing overall rubric scores only modestly. Reports can become stale when the sources they summarize change, yet rerunning the research pipeline can be costly and may unnecessarily rewrite unaffected content. CLAIMWRITER instead converts source additions, removals, and replacements into fact-level edits and maps them to the sections that use those facts. In controlled experiments where a subset of source documents changes, CLAIMWRITER's structured revision achieves the highest rate of incorporating new facts while removing outdated values among the evaluated replay and text-edit methods.

1 INTRODUCTION

Deep-research (DR) systems are agentic pipelines that search across heterogeneous sources, extract evidence, reason over that evidence, and synthesize long-form reports with citations (Venkit et al., 2026). They support information-intensive tasks such as scientific review, policy analysis, market research, and technical investigation. In these settings, the final report is the main artifact presented to the user. Therefore, a useful report must include the task-relevant facts available in the collected evidence, attribute claims to the right sources, and remain maintainable when part of the underlying evidence changes.

Existing DR benchmarks primarily score final reports with holistic rubrics (Du et al., 2026; Wang et al., 2026; Sharma et al., 2026). While useful for overall quality, these scores often hide claim-level failures: reports may omit key facts, cite sources that do not support their claims, or introduce unsupported inferences. In long-form DR reports, such errors can change the conclusions available to readers, yet rubric scores provide little visibility into whether claims are supported, correctly attributed, and complete with respect to the available evidence (Ji et al., 2023; Liu et al., 2023; Venkit et al., 2024).

We address this gap with CLAIMPROBE, a claim-level auditing framework for DR reports. CLAIMPROBE decomposes each report into claims, checks them against the collected source evidence, and measures four dimensions: **hallucination**, **misattribution**, **uncited-but-supported claims**, and **necessary-fact recall**. It then aggregates these judgments into report-level metrics while retaining claim-level diagnostics, so individual failures can be traced to their evidence. Each judgment is produced by an LLM that compares one claim with a short evidence set and is calibrated against human annotations. This localized design follows established LLM-as-judge practice (Zheng et al., 2023).

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.

Applying CLAIMPROBE to Enterprise Deep Research (EDR) (Choubey et al., 2026), NVIDIA AI-Q (NVIDIA, 2025), and OpenDeepResearch (ODR) (LangChain, 2025), we find that their native writers still miss key facts, misattribute evidence, and introduce unsupported claims despite producing fluent reports. These failures point to a limitation of direct report generation from retrieved evidence: facts can lose structure and provenance during writing.

Motivated by this finding, we introduce CLAIMWRITER, a hierarchical evidence-linked writer that replaces only the report-generation module of existing DR systems. We implement two variants: CLAIMWRITER-TD (top-down) routes source-grounded facts into a query-derived outline, whereas CLAIMWRITER-BU (bottom-up) groups the available facts before constructing an outline. Their shared *evidence-linked report representation* connects sources, atomic evidence units, organizational topics, and report sections while retaining provenance. By separating evidence organization from prose generation, CLAIMWRITER can improve recall while preserving links between report claims and their supporting sources.

We evaluate CLAIMWRITER by replacing the native writer in each of these three hosts. Within each host and task, upstream search and evidence collection are held fixed across writer variants. Across the evaluated hosts and judge settings, the ClaimWriter variants reduce hallucination and misattribution and increase necessary-fact recall relative to the corresponding native writers. These gains are largely hidden by rubric-based scores, which move only slightly.

Beyond one-shot generation, CLAIMWRITER also supports report maintenance when source evidence changes. It uses persistent evidence links to map changed source facts to affected sections and is designed to preserve unrelated content. This applies whenever a report must remain current while its underlying sources evolve. In controlled source-change experiments, ClaimWriter most often incorporates changed facts while removing outdated values among the evaluated update methods.

In summary, our main contributions are:

- **Claim-level auditing.** We introduce CLAIMPROBE, a calibrated audit of faithfulness, attribution, support among uncited claims, and evidence coverage in DR reports.
- **Evidence-linked report writing.** We introduce CLAIMWRITER, whose CLAIMWRITER-TD and CLAIMWRITER-BU variants improve measured faithfulness and necessary-fact coverage when writing from fixed retrieved evidence.
- **Source-aware report revision.** We introduce a revision procedure that converts changes in source evidence into fact-level edits and updates only sections linked to changed facts or selected to receive newly added facts.

2 CLAIMPROBE: A CLAIM-LEVEL REPORT AUDITING FRAMEWORK

CLAIMPROBE takes a report and the evidence collected through tool calls, aligns claims and source facts in both directions, and computes four report-level measures. Figure 1 illustrates the two alignment directions; the measures capture failure modes identified in commercial answer engines (Venkit et al., 2024; Liu et al., 2023) and long-form factuality evaluation (Min et al., 2023; Wei et al., 2024).

Hallucination (HALL↓). Does the report make claims that no retrieved source supports?

Misattribution (MIS↓). When the report cites source k, does source k actually support the claim, or is the claim only supported by some other retrieved source?

Uncited-but-supported (UNC). Does the report omit citations for claims that retrieved sources do support? A high UNC together with low HALL and MIS indicates that the writer is grounded but does not always cite.

Necessary fact recall (REC↑). Of the extracted source facts labeled highly relevant to the task, what fraction appear in the report?

2.1 REPORT AUDIT PROCEDURE

CLAIMPROBE first processes the sources and report separately, then performs the two directional comparisons in Figure 1.

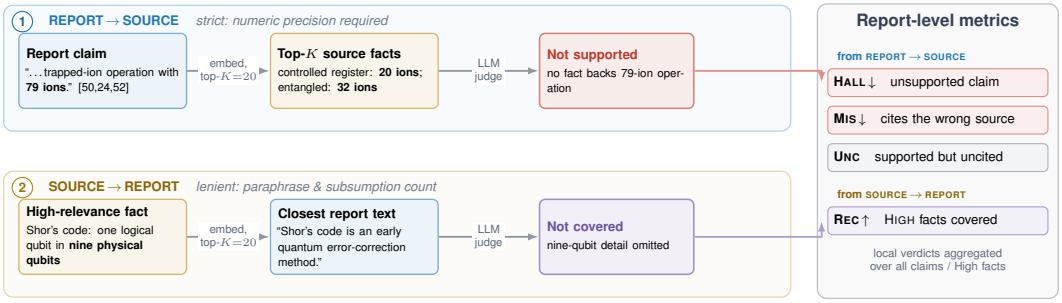

Figure 1: CLAIMPROBE audits a report against its retrieved evidence in two alignment directions, shown here on one concrete example each. Report → source (strict) checks whether each report claim is supported by the top-K retrieved facts and its cited sources; its verdicts yield HALL (unsupported), MIS (cited the wrong source), and UNC (supported but uncited). **Source → report** (lenient) reverses the comparison to test whether each HIGH-relevance source fact appears in the report, yielding REC. In both directions, candidates are retrieved by embedding similarity and scored by a claim-level LLM judge (SUPPORTED/PARTIALLY/NOT); local verdicts are aggregated into the four report-level metrics.

Task	Human-to-human κ	AI/majority	AI/unanimous
Hallucination	0.173	0.299	0.484
Misattribution	0.305	0.570	0.797
Relevance	0.500	0.600	0.743

Table 1: Human and GPT-5.4-mini judge agreement for the three calibrated decisions.

Source processing: converts collected outputs into a standard format and extracts short, self-contained facts. Extraction-time relevance scores are mapped to HIGH, RELATED, or UNRELATED; an optional LLM pass can refine them. Steady-state REC uses only HIGH facts.

Report processing: recovers the reference list and extracts atomic claims section by section, retaining the references cited by each claim.

Alignment: In the report-to-source direction, each claim retrieves the top-K=20 source facts by embedding similarity and an LLM judge assigns SUPPORTED, PARTIALLY SUPPORTED, or NOT SUPPORTED. This direction requires precise factual support, including numeric values. For REC, the same procedure runs from each source fact to report claims; paraphrase, summarization, and subsumption count when the core information is preserved.

Metric computation: HALL counts claims labeled NOT SUPPORTED; its strict variant also counts PARTIALLY SUPPORTED claims. Conversely, REC credits a HIGH source fact when the report fully or partially covers it, while strict REC requires full coverage. MIS checks support in the cited source, and UNC identifies supported claims without citations. Appendix B gives the formal definitions.

Judge calibration: We calibrate GPT-5.4-mini against human annotations. For each of the three decisions, hallucination, misattribution, and relevance, three annotators independently labeled the same stratified pool of 100 items under blind evaluation. Human-to-human agreement is low for hallucination (κ=0.173), modest for misattribution (0.305), and higher for relevance (0.500). Against cases with unanimous human labels, the AI judge reaches κ=0.484, 0.797, and 0.743, respectively; agreement against the human majority is lower but preserves the same ordering (Table 1). The low human agreement for hallucination shows that this decision remains subjective even in the localized setting; we therefore use the calibration as a diagnostic rather than evidence that the judge is definitive.

3 CLAIMWRITER: EVIDENCE-LINKED REPORT WRITING

Overview: Figure 2 summarizes the two CLAIMWRITER variants. Top-down builds a query-derived outline before routing source facts; bottom-up groups the facts before building an evidence-informed outline. Both receive the same query and host-collected pages, then use the query and routed evidence to draft sections, validate citations, and assemble the report. Their internal representation retains

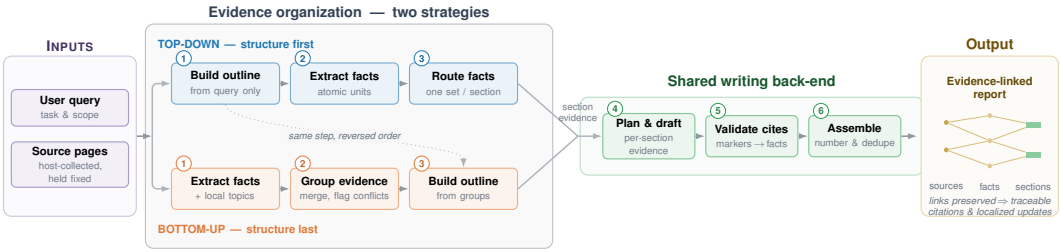

Figure 2: The two CLAIMWRITER variants share one back-end and differ only in when the outline is built. Both receive the same user query and host-collected source pages (INPUTS), then organize evidence by one of two strategies: **top-down** fixes a query-derived outline first and routes extracted facts into it, whereas **bottom-up** extracts and groups facts first and builds the outline from those groups (the "Build outline" step is identical but reversed in order). Both feed per-section evidence into the same **writing back-end** (plan/draft, validate citations, assemble). The retained representation links sources → facts → sections, yielding traceable citations and enabling the localized source-driven updates of Section 3.

links among sources, facts, topics, and sections; bottom-up additionally records fact groups and their hierarchy.

Top-down construction: The default pipeline fixes the requested structure before examining retrieval coverage, so evidence-dense topics do not automatically dominate the report. Its seven steps are:

1. **Normalize the sources.** The writer standardizes host outputs and URLs, removes duplicate pages, and assigns stable page identifiers.
2. **Build an outline from the query.** It proposes and revises topics without seeing the retrieved evidence, making the initial structure independent of retrieval density.
3. **Extract facts from each source.** Each page is processed separately. The extractor produces short, self-contained facts and replaces unclear references with named subjects. It keeps important numbers, dates, and scope. Each fact receives a stable identifier, a link to its source page, and relevance and salience scores. Duplicate facts are then combined.
4. **Assign facts to topics.** The writer sends each retained fact to the one outline topic where it is most useful. A fact that does not support any topic is left unused. As a result, each section receives a small and focused evidence set instead of the full source collection.
5. **Plan and write each section.** For each topic, the writer first makes a short content plan and then drafts the section from its assigned facts. When it uses a fact, it adds a stable marker such as `{{CITE:f0007}}`. If a sentence combines several facts, it includes all matching markers.
6. **Check citations within each section.** A program verifies that every marker names an assigned fact and records assigned facts that were not used.
7. **Assemble the report.** The writer joins sections, merges repeated URLs, numbers references by first use, and deterministically replaces fact markers with numeric citations.

The output is a standard Markdown report, while the retained graph traces citations to sources and records which sections use each fact. This enables section-level checks and source-based updates.

Bottom-up variant: Top-down is simpler and cheaper, but its fixed outline can underuse dense evidence or omit themes not anticipated by the query. As Figure 2 shows, CLAIMWRITER-BU instead groups facts before outlining, allowing available information to affect section depth. It shares normalization, drafting, citation checks, and assembly with top-down; four organization steps replace top-down Steps 2 to 4:

1. **Extract claims and local topics.** Each page yields atomic claims and short topic labels. Claims retain factual detail and citations; labels provide larger units for organizing the report.
2. **Cluster related topics.** CLAIMWRITER-BU first removes irrelevant topics, groups similar page-level topic labels across sources, and then organizes the resulting groups into a hierarchy from broad themes to specific claims. This consolidation treats different names for the same subject as one topic, preventing repeated or fragmented report sections. The original claims and citations remain attached to their groups so the hierarchy does not lose factual detail or provenance.

3. **Find contradictions.** Claims can be compared more reliably after they are grouped by subject. Within each topic group, CLAIMWRITER-BU marks conflicting values, facts, or timelines. It keeps both sides instead of choosing one too early. The final report can therefore describe uncertainty or disagreement between sources.
4. **Build and fill the outline.** The writer uses the query and summaries of the topic groups to create sections, subsections, and writing points. The group summaries keep the outline close to the available evidence. The query keeps common but irrelevant evidence from controlling the report. Each writing point receives the most relevant topic groups. When a group contains one side of a contradiction, the group with the other side is assigned to the same point.

The selected claims then enter the shared writing stages in Figure 2. Because evidence-driven outlines can create uneven sections or weak transitions, CLAIMWRITER-BU reviews drafts for coherence, repetition, and relevance. Its clustering, contradiction analysis, and review increase token cost (Table 7), while both variants preserve links from sections to source evidence.

Updating a report when sources change: These persistent links also support report maintenance. When a source changes, ClaimWriter can identify the facts derived from that source and the report sections that depend on them, instead of treating the report as unstructured text. It converts the source changes into fact-level edits and applies them only to the affected sections in three stages:

1. **Detect fact changes.** For each changed source, an LLM compares the old page, new page, and previously extracted facts. It labels each difference as a *replacement* of an existing fact, a *removal* of a fact no longer supported, or an *addition* not found in the old fact set. Unchanged facts are ignored.
2. **Map changes to sections.** Replaced and removed facts keep their identifiers, which directly locate the sections that used them. Added facts receive new identifiers and are routed into the existing structure. Top-down assigns them to outline topics; bottom-up assigns them to topic groups and then to the best matching sections. This produces one edit bundle per affected section.
3. **Edit and reassemble.** Each affected section is revised once. Replacement keeps the citation marker, removal deletes the unsupported clause and marker, and addition inserts a concise sentence with a new marker. Unaffected sections are reused verbatim, while the editor is instructed to preserve unrelated prose and markers within affected sections. The system then updates its fact records, resolves citations again, and reassembles the report.

This process keeps the original outline and limits generation to sections connected to changed facts. Appendix A.2 gives the exact operations.

4 EXPERIMENTS

We report three experiments. Section 5.1 presents rubric-based report evaluation. Sections 5.2–5.4 present the claim-level faithfulness and coverage evaluation. Section 5.5 evaluates report updating under controlled source changes.

Setup: We conduct our experiments on DeepResearch Bench (Du et al., 2026), which contains 50 English and 50 Chinese deep research tasks; the task scope for each evaluation is specified below. We use three open-source deep research pipelines as host systems for source consolidation: Enterprise Deep Research (EDR) (Choubey et al., 2026), NVIDIA AI-Q (NVIDIA, 2025), and OpenDeepResearch (ODR) (LangChain, 2025). For each pipeline, we replace its original report-writing step with the proposed CLAIMWRITER to generate the final report. We use GPT-5.5 for all generation and host-pipeline components; evaluation judges are specified separately below.

This replacement design isolates report writing from upstream retrieval. Each host performs its own search and evidence collection, so the retrieved pages can differ across EDR, AI-Q, and ODR. We do not treat cross-host score differences as evidence that one retrieval pipeline is better than another. Instead, within each host and task, the native writer and both ClaimWriter variants receive exactly the same host-collected pages. The evidence boundary is fixed, although each writer retains its own internal representation and processing of those pages. ClaimWriter cannot issue new searches, add sources, or change the upstream research trajectory. The controlled comparison therefore asks only how effectively a writer uses a fixed, host-specific evidence boundary: whether it retains task-relevant

facts, avoids claims unsupported by those pages, and attributes claims to the correct retrieved sources. All writers use GPT-5.5 for generation, reducing model-capability differences within these writer comparisons.

Evaluation: For CLAIMPROBE-based evaluation, we use `text-embedding-3-small` as the embedding model and set retrieval to $K=20$ in both directions. We evaluate CLAIMPROBE with two judge settings: GPT-5.4-mini on all 100 tasks (ids 1 to 100) and GPT-5.5 on a 10-task subset (ids 1 to 5 and 51 to 55). Because the task sets differ, their absolute scores are not a controlled judge-only comparison. All results in Table 3 use microaveraging, computed by pooling numerators and denominators across a run's tasks. We also report RACE scores (Du et al., 2026) on all 100 tasks using the original GPT-5.5 judge configuration.

For the dynamic-update evaluation, we use 10 tasks (ids 1 to 5 and 51 to 55). The **5-source update** perturbs five selected sources per task, whereas the **25-source update** perturbs 25 selected sources per task. GPT-5.5 changes one numerical fact in each selected source while leaving the surrounding document unchanged; a separate validation pass checks that the edit is internally consistent with that context. Each edit defines an old and a new target value. We retain targets labeled HIGH or RELATED, because the update experiment asks whether any task-relevant changed information propagates, whereas steady-state REC deliberately uses only HIGH facts. The resulting evaluation sets contain 42 and 204 facts. We define *end-to-end update success* as the number of retained perturbations for which the regenerated report contains the new value and no longer contains the old value, divided by the total number of retained perturbations. An LLM evaluator determines whether the revised report expresses the new value and omits the old value, including when the values appear in paraphrased prose or tables. The denominator includes every retained perturbation, even when its old value was absent from the initial report. The metric therefore combines initial coverage with update success; it is not conditional replacement accuracy. We report this rate together with token cost and post-update RACE scores.

Baseline Systems: For each host and task, we first obtain the native report and the evidence collected by the host pipeline. We then hold that evidence fixed and replace only the report-writing stage. The **Baseline** is the host's native report writer. We compare the host writer with CLAIMWRITER-TD and CLAIMWRITER-BU, both described in Section 3. They receive the same retrieved pages and use the same fact-linked section-writing and citation-assembly design. They differ in how evidence is represented and organized before writing: CLAIMWRITER-TD routes extracted facts into a query-derived outline, whereas CLAIMWRITER-BU groups extracted claims before building the outline. For report updating, all systems use the same AI-Q-collected sources, but they begin from reports produced by different writers. **AI-Q native replay** updates the native AI-Q report by replacing changed source contents in the saved execution trace, reusing unaffected cached nodes, and re-invoking downstream nodes reached by the replay. **Text-edit** operates only on source and report text. It first compares each original and changed source to extract value replacements, removals, and additions. It then gives the complete change list to every report section and asks the model to decide which changes are relevant and rewrite that section while preserving unrelated prose and citations. We apply text-edit to the native AI-Q, CLAIMWRITER-TD, and CLAIMWRITER-BU reports. **Fact-edit** instead uses ClaimWriter's stored fact-to-section links to route each change directly to affected sections; we apply it to the corresponding CLAIMWRITER-TD and CLAIMWRITER-BU reports.

5 RESULTS

5.1 HOLISTIC RUBRIC SCORES SHOW A MODEST DOWNWARD TREND

Table 2 shows a modest but consistent downward trend in holistic rubric scores after replacing the native writer. Overall RACE decreases by 0.006 to 0.018 across hosts and ClaimWriter variants. The decline is concentrated in readability, which falls by 0.038 to 0.059, while comprehensiveness and insight generally remain within 0.013 of the native writer. Bottom-up recovers some comprehensiveness and insight relative to top-down, but at substantially higher generation cost (Appendix E). ClaimWriter therefore does not improve the holistic rubric score; instead, it preserves most of that score while targeting evidence faithfulness and coverage, which we evaluate directly in the next section.

Host + writer	comp.	insi.	inst.	read.	**ov.**
EDR	**0.556**	0.548	**0.543**	**0.510**	**0.544**
EDR + CLAIMWRITER-TD	0.553	0.545	0.537	0.470	0.536
EDR + CLAIMWRITER-BU	0.554	**0.550**	0.539	0.466	0.538
AI-Q	**0.558**	**0.557**	**0.554**	**0.528**	**0.553**
AI-Q + CLAIMWRITER-TD	0.550	0.544	0.536	0.473	0.535
AI-Q + CLAIMWRITER-BU	0.556	0.552	0.541	0.469	0.540
ODR	**0.546**	0.539	**0.544**	**0.523**	**0.540**
ODR + CLAIMWRITER-TD	0.543	0.534	0.534	0.485	0.530
ODR + CLAIMWRITER-BU	**0.546**	**0.542**	0.536	0.472	0.533

Table 2: RACE scores (n=100, judge GPT-5.5). Columns are comprehensiveness, insight, instruction-following, readability, and overall (higher is better). Within each host and metric, green marks the highest score, yellow the middle score, and red the lowest; tied highest scores are both green.

		GPT-5.5 (n=10)						**GPT-5.4-mini** (n=100)					
Host	Writer	HALL↓	HALL_{str}↓	MIS↓	REC↑	REC_{str}↑	UNC	HALL↓	HALL_{str}↓	MIS↓	REC↑	REC_{str}↑	UNC
AI-Q	Baseline	14.84	78.35	24.68	36.08	7.85	83.23	23.27	73.51	22.31	51.35	20.59	73.83
	+CLAIMWRITER-BU	3.41	**36.80**	5.29	56.31	28.46	88.92	**5.08**	**28.09**	4.13	**68.71**	46.59	79.61
	+CLAIMWRITER-TD	**3.37**	39.02	**5.10**	**56.99**	**29.97**	88.46	5.20	28.49	**3.62**	68.29	**47.75**	80.61
ODR	Baseline	23.80	75.67	7.67	33.61	9.21	66.68	27.12	67.08	5.92	59.34	32.82	63.60
	+CLAIMWRITER-BU	**5.16**	**42.34**	**2.23**	**63.79**	**43.89**	78.06	**6.78**	**31.16**	**2.28**	**80.48**	**67.79**	73.38
	+CLAIMWRITER-TD	5.92	48.90	2.38	57.82	36.58	78.04	8.78	35.87	2.50	75.93	61.13	72.60
EDR	Baseline	12.09	61.86	24.23	21.12	3.92	85.11	15.89	54.05	18.94	36.83	13.67	77.75
	+CLAIMWRITER-BU	**3.91**	**37.67**	6.89	**31.72**	11.73	91.14	**4.91**	28.56	5.88	43.99	22.69	82.71
	+CLAIMWRITER-TD	4.65	39.31	**6.77**	31.56	**11.79**	86.38	5.02	**28.55**	**5.43**	**45.85**	**24.48**	83.57

Table 3: CLAIMPROBE results by host, writer, and judge setting. All values are percentages, micro-averaged over pooled (numerator, denominator) pairs. GPT-5.5 uses 10 tasks and GPT-5.4-mini uses 100, so absolute values across the two blocks are not a controlled judge comparison. Within each host and metric, green marks the best observed result, yellow an intermediate result, and red the weakest; UNC is contextual and left uncolored.

5.2 NATIVE REPORT WRITERS OMIT EVIDENCE AND INTRODUCE UNSUPPORTED CLAIMS

We next ask whether the native report writers use their retrieved evidence reliably. Under the 100-task CLAIMPROBE evaluation, their reports have hallucination rates of 15.89% to 27.12%, misattribution rates of 5.92% to 22.31%, and necessary-fact recall of only 36.83% to 59.34% (Table 3). The 10-task judge setting shows the same broad pattern: every native writer has the highest hallucination and the weakest recall results within its host block. Thus, the problem is not confined to one DR system or judge configuration. Native reports contain unsupported claims, sometimes cite the wrong retrieved evidence, and omit a substantial share of high-relevance facts. These failures are largely hidden by the comparatively close holistic scores in Table 2.

5.3 CLAIMWRITER SUBSTANTIALLY REDUCES CLAIM-LEVEL FAILURES

Replacing the native writer with a ClaimWriter variant lowers hallucination by 7 to 21 percentage points and raises necessary-fact recall by 7 to 30 points across the evaluated host and judge configurations (Table 3). Misattribution also decreases substantially in most settings. For example, on EDR with the 100-task judge, CLAIMWRITER-TD reduces HALL from 15.89% to 5.02% and MIS from 18.94% to 5.43%, while increasing REC from 36.83% to 45.85%. CLAIMWRITER-TD and CLAIMWRITER-BU trade the lead across individual metrics, but both consistently outperform the native writer on the primary faithfulness and coverage measures. The two judge settings use different task sets, so their absolute values should not be compared directly; the robustness analysis below separately tests judge choice.

Support among uncited claims: The uncited-but-supported rate (UNC) is high in every setting, ranging from 63.60% to 91.14%, and it often rises with ClaimWriter. Its denominator is the set of uncited claims: UNC measures the fraction of those claims that the retrieved evidence nevertheless supports. It is therefore not a citation-completeness measure and does not reveal what fraction of all

Initial report	Update method	**5 changed sources/task** (n=42 relevant facts) In Tok	Out Tok	Total Tok	E2E success [%]	**25 changed sources/task** (n=204 relevant facts) In Tok	Out Tok	Total Tok	E2E success [%]
AI-Q native	native replay	369K	159K	528K	**0.0**	815K	354K	1.17M	**0.5**
AI-Q native	text-edit	44K	20K	64K	16.7	119K	21K	140K	6.9
CLAIMWRITER-TD	text-edit	88K	68K	156K	40.5	152K	70K	222K	30.9
CLAIMWRITER-TD	fact-edit	49K	34K	83K	**47.6**	146K	72K	218K	39.7
CLAIMWRITER-BU	text-edit	110K	91K	201K	42.9	176K	93K	268K	32.4
CLAIMWRITER-BU	fact-edit	100K	78K	178K	45.2	188K	100K	288K	**41.2**

Table 4: Dynamic-update cost and end-to-end update success over 10 tasks using AI-Q-collected sources. Input, output, and total tokens are per-report averages. End-to-end (E2E) success is the percentage of retained HIGH+RELATED perturbations for which the regenerated report contains the new value and not the old value; its denominator includes facts absent from the initial report. Green marks the best success rate, yellow intermediate rates, red weak rates, and dark red a near-floor rate; token columns are uncolored.

supported claims lack citations. Its increase alongside lower HALL and MIS and higher REC indicates that many remaining uncited claims are grounded; it does not reverse the primary faithfulness and coverage gains.

5.4 HOLISTIC QUALITY AND SOURCE FAITHFULNESS CAN DIVERGE

The RACE result in Table 2 and the CLAIMPROBE result in Table 3 show that holistic rubric scores and source faithfulness are partly separable. A writer can keep the holistic evaluation score close to the baseline while changing the claim-to-source relationship substantially. In other words, the broad report-quality measures are useful but insufficient in practice: they should be paired with a claim-level audit that checks whether the report cites the right sources and retains the necessary facts.

Why claim-level judging is better scoped: CLAIMPROBE also uses an LLM judge, but each decision is local. A HALL or MIS verdict compares one report claim against a top-K=20 evidence shortlist; a REC verdict asks whether one HIGH-relevance source fact is paraphrased or subsumed in the report. Since the judge never evaluates either artifact end to end, page-level confounds such as long-prose anchoring, length bias, and presentation effects are reduced. This matches the observation from Zheng et al. (2023) that LLM-as-a-judge is more reliable under tightly scoped prompts, and our human study (Section 2) confirms higher agreement in this restricted setting.

Robustness to judge choice: The main audit uses a GPT-family judge, which could raise concerns about family-specific preferences because GPT-5.5 also writes the reports. We therefore repeat the audit with Gemini-3.5-flash on the 10-task subset while holding reports, sources, extraction, and retrieval fixed. The absolute values change, but every ClaimWriter variant retains its improvement over the corresponding baseline on the reported faithfulness and recall measures. For example, on AI-Q, replacing the native writer with CLAIMWRITER-BU reduces HALL from 57.14% to 19.81%, reduces MIS from 15.89% to 0.86%, and raises REC from 15.97% to 41.22%. Appendix C reports the full factorial table.

5.5 EVIDENCE LINKS IMPROVE REPORT UPDATING AT LOWER COST

Next, we evaluate whether a report incorporates controlled changes to its source documents. For example, if a source value changes from 2024 to 2025, an update succeeds only when the revised report contains 2025 and no longer contains 2024. Our end-to-end update-success rate uses all retained task-relevant perturbations as its denominator, including facts absent from the original report. It therefore captures both whether the initial writer covered a fact and whether the updater propagated its new value. Table 4 reports this rate together with average update-time token cost per report.

Effect of the update mechanism. The controlled comparisons hold the initial report fixed and vary only the updater. On the native AI-Q report, text-edit improves end-to-end success over native replay by 16.7 percentage points in the 5-source update (16.7% versus 0.0%) and 6.4 points in the 25-source update (6.9% versus 0.5%). For CLAIMWRITER-TD reports, fact-edit improves over text-edit by 7.1 points in the 5-source update (47.6% versus 40.5%) and 8.8 points in the 25-source update (39.7% versus 30.9%). For CLAIMWRITER-BU reports, the corresponding improvements are 2.3 and 8.8

points. These paired comparisons show that stored fact-to-section links improve updating beyond an unstructured textual diff.

Effect of the initial report. Comparisons across report types reflect both initial evidence coverage and updater behavior. Under text-edit, CLAIMWRITER-TD reaches 40.5% and 30.9%, and CLAIMWRITER-BU reaches 42.9% and 32.4%, compared with 16.7% and 6.9% for the native AI-Q report. These differences cannot be attributed solely to the updater because ClaimWriter reports may contain more perturbed facts before updating. CLAIMWRITER-TD fact-edit leads the 5-source update, while CLAIMWRITER-BU fact-edit leads the 25-source update; nevertheless, every end-to-end success rate remains below 50%.

Update cost. Fact-edit is also substantially cheaper than native replay. CLAIMWRITER-TD fact-edit uses 83K tokens per report in the 5-source update, compared with 528K for replay, a reduction of 84.3%. In the 25-source update it uses 218K rather than 1.17M tokens, a reduction of 81.4%. CLAIMWRITER-BU fact-edit uses 178K and 288K tokens, respectively. Thus, both ClaimWriter variants update more changed facts while using fewer update-time tokens than native replay. ClaimWriter's structured representation has an up-front cost but makes later updates cheaper; Appendix E reports first-run token costs and a conservative break-even analysis.

6 RELATED WORK

Web-grounded long-form generation: WebGPT and GopherCite established search-grounded generation (Nakano et al., 2021; Menick et al., 2022); ASQA and QAMPARI expose multi-fact coverage failures (Stelmakh et al., 2022; Amouyal et al., 2023); and STORM organizes retrieved information before writing (Shao et al., 2024). CLAIMWRITER instead preserves atomic-fact provenance over evidence collected by a DR host and supports source-driven updates.

Deep-research systems and evaluation: DR systems include proprietary offerings (Citron, 2024; OpenAI, 2025) and open implementations (Choubey et al., 2026; NVIDIA, 2025; LangChain, 2025). Benchmarks such as DeepResearch Bench, LiveResearch Bench, ResearchRubrics, DeepResearch Bench II, DRBench, and GAIA score reports or complete trajectories (Du et al., 2026; Wang et al., 2026; Sharma et al., 2026; Li et al., 2026; Abaskohi et al., 2026; Mialon et al., 2024). Because these evaluations combine retrieval and writing, they do not isolate where failures arise. Our writer swap holds each host's evidence fixed and tests how the writer covers, supports, and attributes it.

Claim-level factuality and attribution: FActScore, SAFE, and VeriScore decompose long-form generations into verifiable claims (Min et al., 2023; Wei et al., 2024; Song et al., 2024); AIS, ALCE, attributed QA, and RAGAS study attribution and citation quality (Rashkin et al., 2023; Gao et al., 2023b; Bohnet et al., 2022; Es et al., 2024). Audits nevertheless find unsupported claims and citation errors in fluent search-generated answers (Liu et al., 2023; Venkit et al., 2024; 2026). CLAIMPROBE treats the retrieved corpus as the evidence boundary and measures both report-to-source failures and source-to-report coverage, isolating how a writer uses evidence.

Post-hoc correction and report maintenance: RARR retrieves attribution and revises unsupported passages while preserving the original response (Gao et al., 2023a), but must recover evidence links from unstructured prose. CLAIMWRITER creates these links during generation, so source changes can become fact additions, removals, or replacements routed to affected sections. We thus study repeated maintenance of structured reports rather than one-time repair.

7 CONCLUSION

We introduce CLAIMPROBE, a claim-level audit for DR reports, and CLAIMWRITER, an evidence-linked DR report writer. Across three DR hosts, replacing only their writer largely preserves rubric-based performance while improving measured claim-level faithfulness and recall. The same representation supports localized report revision under source drift: on AI-Q-collected sources, CLAIMWRITER-TD and CLAIMWRITER-BU fact-edit achieve the highest end-to-end update-success rates in the 5-source and 25-source updates, respectively. All end-to-end success rates remain below 50%. These results motivate further work on report writers and update systems that preserve evidence structure, attribution, and maintainability.

AI USE STATEMENT

We used generative AI tools for writing assistance, including prose editing and LaTeX formatting, and for assistance with implementation, evaluation, and experiment-orchestration scripts. The research questions, method design, experimental protocol, interpretation, and conclusions are the authors' own. All AI-assisted code was reviewed and tested by the authors, and all reported numbers were produced by the experimental pipeline and checked against the underlying evaluation outputs and logs. Generative models also form explicit parts of the proposed method and evaluation: GPT-5.5 is used by the report-writing and controlled source-perturbation pipelines; GPT-5.4-mini and GPT-5.5 serve as CLAIMPROBE judges in the reported settings; and Gemini-3.5-flash is used for the independent judge-robustness analysis. These model uses are inference-only and do not constitute model training. We take responsibility for the final text, claims, code, and experimental artifacts, including material produced with generative-AI assistance.

ETHICS STATEMENT

This work studies the generation, auditing, and maintenance of deep-research reports using public benchmark tasks and retrieved web sources. It does not train or release a new foundation model, deploy a system to end users, or intentionally collect private user data. The human study is limited to annotating generated claims and their supporting evidence for judge calibration; no personally identifiable participant data is reported. More faithful citation and update mechanisms can improve the reliability of research reports, but they do not eliminate hallucination, source bias, outdated evidence, or misleading attribution. Users should therefore treat generated reports as decision support rather than verified ground truth, especially in high-stakes domains. We take responsibility for the study design, released artifacts, and reported conclusions.

REPRODUCIBILITY STATEMENT

Section 3 specifies the ClaimWriter pipelines, Section 2 defines ClaimProbe, and Section 4 describes the datasets, systems, models, evaluation subsets, and source-perturbation protocol. Appendix A documents implementation and update operations; Appendices B and F provide metric definitions, judge directions, and extraction and update schemas; and Appendix E reports the available token accounting. The source code, full prompts, pipeline configurations, evaluation scripts, and benchmark will be released upon publication.

REFERENCES

Amirhossein Abaskohi, Tianyi Chen, Miguel Muñoz-Mármol, Curtis Fox, Amrutha Varshini Ramesh, Étienne Marcotte, Xing Han Lù, Nicolas Chapados, Spandana Gella, Christopher Pal, Alexandre Drouin, and Issam H. Laradji. Drbench: A realistic benchmark for enterprise deep research. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/0ba520d93c3df592c83a611961314c98-Abstract-Conference.html.

Samuel Amouyal, Tomer Wolfson, Ohad Rubin, Ori Yoran, Jonathan Herzig, and Jonathan Berant. QAMPARI: A benchmark for open-domain questions with many answers. In *Proceedings of the Third Workshop on Natural Language Generation, Evaluation, and Metrics (GEM)*, pp. 97–110. Association for Computational Linguistics, 2023. URL https://aclanthology.org/2023.gem-1.9/.

Bernd Bohnet, Vinh Q. Tran, Pat Verga, Roee Aharoni, Daniel Andor, Livio Baldini Soares, Massimiliano Ciaramita, Jacob Eisenstein, Kuzman Ganchev, Jonathan Herzig, Kai Hui, Tom Kwiatkowski, Ji Ma, Jianmo Ni, Lierni Sestorain Saralegui, Tal Schuster, William W. Cohen, Michael Collins, Dipanjan Das, Donald Metzler, Slav Petrov, and Kellie Webster. Attributed question answering: Evaluation and modeling for attributed large language models. *arXiv preprint arXiv:2212.08037*, 2022.

Prafulla Kumar Choubey, Kung-Hsiang Huang, Pranav Narayanan Venkit, Jiaxin Zhang, Vaibhav Vats, Yu Li, Xiangyu Peng, and Chien-Sheng Wu. Don't stop early: Scalable enterprise deep

research with controlled information flow and evidence-aware termination. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track)*, pp. 1699–1713. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-industry.116. URL https://aclanthology.org/2026.acl-industry.116/.

Dave Citron. Try deep research and our new experimental model in gemini, your AI assistant. Google blog post, December 2024. URL https://blog.google/products-and-platforms/products/gemini/google-gemini-deep-research/.

Mingxuan Du, Benfeng Xu, Chiwei Zhu, Licheng Zhang, Xiaorui Wang, and Zhendong Mao. DeepResearch Bench: A comprehensive benchmark for deep research agents. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=hQ0K2Hhq7H.

Shahul Es, Jithin James, Luis Espinosa-Anke, and Steven Schockaert. RAGAs: Automated evaluation of retrieval augmented generation. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pp. 150–158. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.eacl-demo.16. URL https://aclanthology.org/2024.eacl-demo.16/.

Luyu Gao, Zhuyun Dai, Panupong Pasupat, Anthony Chen, Arun Tejasvi Chaganty, Yicheng Fan, Vincent Zhao, Ni Lao, Hongrae Lee, Da-Cheng Juan, and Kelvin Guu. RARR: Researching and revising what language models say, using language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pp. 16477–16508. Association for Computational Linguistics, 2023a. doi: 10.18653/v1/2023.acl-long.910. URL https://aclanthology.org/2023.acl-long.910/.

Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. Enabling large language models to generate text with citations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 6465–6488. Association for Computational Linguistics, 2023b. doi: 10.18653/v1/2023.emnlp-main.398. URL https://aclanthology.org/2023.emnlp-main.398/.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM computing surveys*, 55(12):1–38, 2023. doi: 10.1145/3571730. URL https://doi.org/10.1145/3571730.

LangChain. Open deep research. GitHub repository, 2025. URL https://github.com/langchain-ai/open_deep_research.

Ruizhe Li, Mingxuan Du, Benfeng Xu, Chiwei Zhu, Xiaorui Wang, and Zhendong Mao. Deepresearch bench ii: Diagnosing deep research agents via rubrics from expert report, 2026. URL https://arxiv.org/abs/2601.08536.

Nelson F Liu, Tianyi Zhang, and Percy Liang. Evaluating verifiability in generative search engines. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 7001–7025. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.findings-emnlp.467. URL https://aclanthology.org/2023.findings-emnlp.467/.

Jacob Menick, Maja Trebacz, Vladimir Mikulik, John Aslanides, Francis Song, Martin Chadwick, Mia Glaese, Susannah Young, Lucy Campbell-Gillingham, Geoffrey Irving, and Nat McAleese. Teaching language models to support answers with verified quotes. *arXiv preprint arXiv:2203.11147*, 2022. URL https://arxiv.org/abs/2203.11147.

Grégoire Mialon, Clémentine Fourrier, Craig Swift, Thomas Wolf, Yann LeCun, and Thomas Scialom. GAIA: A benchmark for general AI assistants. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/25ae35b5b1738d80f1f03a8713e405ec-Abstract-Conference.html.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 12076–12100. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.741. URL https://aclanthology.org/2023.emnlp-main.741/.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Karl Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin Chess, and John Schulman. WebGPT: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*, 2021. URL https://arxiv.org/abs/2112.09332.

NVIDIA. NVIDIA AI-Q Blueprint. GitHub repository, 2025. URL https://github.com/NVIDIA-AI-Blueprints/aiq.

OpenAI. Introducing deep research. OpenAI blog post, 2025. URL https://openai.com/index/introducing-deep-research/.

Hannah Rashkin, Vitaly Nikolaev, Matthew Lamm, Lora Aroyo, Michael Collins, Dipanjan Das, Slav Petrov, Gaurav Singh Tomar, Iulia Turc, and David Reitter. Measuring attribution in natural language generation models. *Computational Linguistics*, 49(4):777–840, 2023. doi: 10.1162/coli_a_00486. URL https://aclanthology.org/2023.cl-4.2/.

Yijia Shao, Yucheng Jiang, Theodore Kanell, Peter Xu, Omar Khattab, and Monica Lam. Assisting in writing Wikipedia-like articles from scratch with large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 6252–6278. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.naacl-long.347. URL https://aclanthology.org/2024.naacl-long.347/.

Manasi Sharma, Chen Bo Calvin Zhang, Chaithanya Bandi, Clinton Wang, Ankit Aich, Huy Nghiem, Tahseen Rabbani, Ye Htet, Brian Jang, Sumana Basu, Aishwarya Balwani, Denis Peskoff, Marcos Ayestaran, Sean Hendryx, Bradley Kenstler, and Bing Liu. ResearchRubrics: A benchmark of prompts and rubrics for evaluating deep research agents. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2511.07685.

Yixiao Song, Yekyung Kim, and Mohit Iyyer. VeriScore: Evaluating the factuality of verifiable claims in long-form text generation. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 9447–9474. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-emnlp.552. URL https://aclanthology.org/2024.findings-emnlp.552/.

Ivan Stelmakh, Yi Luan, Bhuwan Dhingra, and Ming-Wei Chang. ASQA: Factoid questions meet long-form answers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 8273–8288. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.emnlp-main.566. URL https://aclanthology.org/2022.emnlp-main.566/.

Pranav Narayanan Venkit, Philippe Laban, Yilun Zhou, Yixin Mao, and Chien-Sheng Wu. Search engines in an ai era: The false promise of factual and verifiable source-cited responses. *arXiv preprint arXiv:2410.22349*, 2024.

Pranav Narayanan Venkit, Philippe Laban, Yilun Zhou, Kung-Hsiang Huang, Yixin Mao, and Chien-Sheng Wu. DeepTRACE: Auditing deep research AI systems for tracking reliability across citations and evidence. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/ad08767706825033b99122332293033d-Abstract-Conference.html.

Jiayu Wang, Yifei Ming, Riya Dulepet, Qinglin Chen, Austin Xu, Zixuan Ke, Frederic Sala, Aws Albarghouthi, Caiming Xiong, and Shafiq Joty. LiveResearchBench: A live benchmark for user-centric deep research in the wild. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/998f8d8ff2697da2eae5f87143668754-Abstract-Conference.html.

Jerry Wei, Chengrun Yang, Xinying Song, Yifeng Lu, Nathan Hu, Jie Huang, Dustin Tran, Daiyi Peng, Ruibo Liu, Da Huang, Cosmo Du, and Quoc V. Le. Long-form factuality in large language models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 80756–80827, 2024. doi: 10.52202/079017-2567.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. *Advances in neural information processing systems*, 36:46595–46623, 2023.

A ClaimWriter implementation details

A.1 Fact extraction, routing, and drafting

Each source page is processed independently and in parallel. The extractor returns at most 25 self-contained facts per page and preserves named subjects, exact quantities, time references, and scope. It also predicts task relevance and salience, which are used to prioritize facts before writing. The shared extraction schema and quality gates appear in Appendix F. After cross-page deduplication, routing assigns each retained fact to one outline topic rather than broadcasting the same fact to every section.

For each section, the writer creates a short plan of two to five content beats and then drafts against only the facts assigned to that section. Citations remain fact-id markers throughout this step. A programmatic annotation pass resolves valid markers against the section's evidence and records facts supplied but never cited. Assembly then deduplicates source URLs, assigns reference numbers in first-use order, replaces resolved fact markers with source numbers, and emits the report and reference list. The bottom-up variant adds an LLM coherence review and selectively regenerates failing sections. Prompts and decoding configurations will be released with the code; the extraction and update schemas used in our experiments are reproduced in Appendix F.

A.2 Fact-diff operations

For each changed page, the updater compares the new page with its previous facts. It applies three operations:

- **Replace:** update the fact text while preserving its identifier and topic assignment, then revise sections containing its marker.
- **Remove:** remove the obsolete fact and revise each section containing its marker so that both the unsupported clause and marker disappear.
- **Add:** extract a new fact, allocate a new identifier, route it to the fixed outline, and revise the receiving section to incorporate it when relevant.

Each revised section passes through the same citation annotation and finalization steps as first-run generation. The updater operates against the existing outline; changes requiring a different information architecture are outside the localized-update setting and require fresh report construction.

B ClaimProbe pipeline details

Extract. Tool logs become a normalized `sources` table with URL canonicalization (scheme, `www` prefix, trailing slash, and fragments removed) and a boilerplate stripper. The report's reference list is parsed by three regular expressions (canonical `[N] url`, numbered Markdown `N. [Title](url)`, and descriptive `[N] description: url`) to tolerate real-world citation conventions. The report is segmented twice: `report_units.json` (sentence units, table summaries, contextualized table-cell atoms) for Rec, and `report_claims.json` (claims) for Hall, Mis, Unc. Source contents are segmented into source facts by an LLM extractor instructed to make facts self-sufficient, replace pronouns with entity names, and drop boilerplate.

Relevance. Each source fact receives a label in {High, Related, Unrelated} conditioned on the task statement. High facts directly answer the task or provide a central statistic the report is

expected to use; RELATED facts provide background; UNRELATED facts are off-topic. UNRELATED facts are excluded from the embedding index.

Alignment. For each query item we retrieve the top-K=20 candidates and ask a direction-aware claim-level judge for a strict JSON verdict in {S, PS, NS} with a 1-to-5 confidence score and a 20-word justification. The report-to-source prompt is strict (numeric precision required); the source-to-report prompt is lenient (paraphrase and subsumption count as coverage).

Metrics. With $\mathcal{C}$ the set of report claims and $\mathcal{F}_{\text{high}}$ the HIGH source-fact set,

$$\begin{aligned}
\text{HALL} &= |\{c : \texttt{r2s}(c){=}\text{NS}\}| \,/\, |\mathcal{C}| \\
\text{HALL}_{\text{str}} &= |\{c : \texttt{r2s}(c){\neq}\text{S}\}| \,/\, |\mathcal{C}| \\
\text{MIS} &= |\{c{\in}\mathcal{C}_{\text{cited}} : \neg\text{by_cited}(c) \wedge \text{by_other}(c)\}| \,/\, |\mathcal{C}_{\text{cited}}| \\
\text{UNC} &= |\{c{\in}\mathcal{C}_{\text{uncited}} : \text{supported}(c)\}| \,/\, |\mathcal{C}_{\text{uncited}}| \\
\text{REC} &= |\{f{\in}\mathcal{F}_{\text{high}} : \texttt{s2r}(f){\in}\{\text{S}, \text{PS}\}\}| \,/\, |\mathcal{F}_{\text{high}}|.
\end{aligned}$$

A site-prefix fallback (`netloc + first two path segments`) avoids inflating MIS on government and statistical mega-domains.

C ROBUSTNESS TO THE JUDGE MODEL

Table 5 gives the complete independent-judge results for the 10-task subset summarized in Section 5.4. Reports, sources, extraction, and retrieval are held fixed; only the judge is changed to Gemini-3.5-flash on tasks 1 to 5 and 51 to 55.

		Gemini-3.5-flash (n=10)					
Host	Writer	HALL↓	HALL_{str}↓	MIS↓	REC↑	REC_{str}↑	UNC
AI-Q	Baseline	57.14	83.69	15.89	15.97	5.43	28.72
	+CLAIMWRITER-BU	**19.81**	**49.43**	**0.86**	**41.22**	**27.14**	19.05
	+CLAIMWRITER	23.03	54.92	0.99	39.71	25.89	22.86
ODR	Baseline	58.72	81.94	2.51	22.49	9.25	23.25
	+CLAIMWRITER-BU	**23.50**	**54.96**	**0.27**	**59.64**	**46.72**	35.71
	+CLAIMWRITER	29.08	58.45	0.30	52.35	40.42	25.21
EDR	Baseline	45.24	71.19	12.86	9.65	3.32	26.20
	+CLAIMWRITER-BU	**20.57**	**48.21**	**1.15**	**17.23**	**9.91**	26.67
	+CLAIMWRITER	25.27	55.51	1.39	17.10	9.80	21.74

Table 5: CLAIMPROBE audit with the Gemini-3.5-flash judge (n=10; ids 1 to 5 and 51 to 55). All values are percentages, micro-averaged over pooled (numerator, denominator) pairs. Green cells are the best writer in each host block; UNC is contextual and left uncolored.

D RACE UNDER SOURCE PERTURBATION

Table 6 reports holistic quality after regeneration for the same 10 tasks used in the dynamic-update evaluation. Updates starting from the native AI-Q report retain the highest RACE scores, close to the unperturbed AI-Q score. This should not be interpreted as successful updating: Table 4 shows that these methods have low end-to-end update success. For the CLAIMWRITER-BU report in the 5-source update, text-edit preserves nearly the same overall RACE score as fact-edit (0.5426 versus 0.5467), although its end-to-end success is lower (42.9% versus 45.2%). More generally, the ClaimWriter-based reports have lower readability scores and higher update-success rates, but this correlation does not establish that the updates caused the readability difference.

E WRITING-ONLY GENERATION COST

Table 7 reports average per-report token usage for the CLAIMWRITER-TD and CLAIMWRITER-BU pipelines over the 100 Deep Research Bench tasks. It excludes native host-writing costs because research and writing are not separated consistently across the three host pipelines.

Initial report	Update method	5-source update					25-source update				
		Comp.	Insi.	Inst.	Read.	**Ov.**	Comp.	Insi.	Inst.	Read.	**Ov.**
AI-Q native	native replay	0.5584	0.5559	**0.5594**	**0.5326**	**0.5545**	0.5549	**0.5571**	0.5530	**0.5311**	0.5522
AI-Q native	text-edit	0.5558	0.5543	0.5581	0.5312	0.5527	0.5569	0.5554	**0.5551**	0.5280	**0.5524**
CLAIMWRITER-TD	text-edit	0.5425	0.5415	0.5375	0.4721	0.5327	0.5555	0.5448	0.5471	0.4693	0.5390
CLAIMWRITER-TD	fact-edit	0.5539	0.5413	0.5410	0.4758	0.5370	0.5527	0.5389	0.5378	0.4696	0.5341
CLAIMWRITER-BU	text-edit	0.5609	0.5525	0.5396	0.4741	0.5426	0.5550	0.5513	0.5375	0.4667	0.5392
CLAIMWRITER-BU	fact-edit	**0.5618**	**0.5583**	0.5440	0.4800	0.5467	**0.5605**	0.5520	0.5408	0.4667	0.5418

Table 6: RACE after perturbing AI-Q-collected sources on 10 tasks (ids 1 to 5 and 51 to 55; judge GPT-5.5). Within each condition and metric, green marks the highest score, yellow an intermediate score, and red the lowest; ties for the lowest score are both red. High RACE preservation does not imply successful updating; end-to-end update-success rates are reported in Table 4.

Host	Writer	Sources/report	Input	Output	Total
AI-Q	CLAIMWRITER-TD	306.2	1.108M	0.472M	1.580M
AI-Q	CLAIMWRITER-BU	306.2	2.861M	1.187M	4.048M
ODR	CLAIMWRITER-TD	73.7	0.297M	0.143M	0.440M
ODR	CLAIMWRITER-BU	73.7	1.146M	0.520M	1.667M
EDR	CLAIMWRITER-TD	1,125.2	3.166M	1.100M	4.266M
EDR	CLAIMWRITER-BU	1,125.2	6.304M	2.531M	8.835M

Table 7: Average first-run ClaimWriter token cost per report over 100 tasks. Sources/report is the mean number of deduplicated source URLs supplied by the host. Native host-writing cost is omitted because its calls are entangled with the host's research pipeline and do not admit a consistent writing-only boundary.

The first-run cost depends strongly on the evidence volume supplied by the host: ODR provides about 74 source pages per report, AI-Q 306, and EDR 1,125; the corresponding CLAIMWRITER-TD costs are 0.44M, 1.58M, and 4.27M tokens. The relationship is not strictly proportional because source length, fact density, and report structure also vary. CLAIMWRITER-BU costs more at every scale because it additionally clusters evidence, builds a hierarchy, checks contradictions, and reviews the draft.

For AI-Q, a conservative break-even calculation can charge the *entire* first-run ClaimWriter cost against the token savings over native replay. Under repeated updates of the same scale, CLAIMWRITER-TD recovers its 1.58M-token initial cost after approximately four 5-source updates or two 25-source updates; CLAIMWRITER-BU recovers its 4.05M-token cost after approximately 12 or five updates, respectively. These are upper bounds on the amortization point because ClaimWriter replaces some native writing that would otherwise incur its own cost. We omit native first-run writing tokens: in the evaluated systems, report production is distributed across research, compression, synthesis, and writing calls, so a comparable writing-only boundary cannot be recovered consistently. The break-even estimate therefore uses only directly measured ClaimWriter generation and update costs and should be read as a token-cost comparison, not an end-to-end compute estimate.

F JUDGE AND EXTRACTOR PROMPTS

We reproduce the system prompts used by the CLAIMPROBE judge and the extractors. Long extractor prompts are shown as their core skeleton plus the hard-rule list; the complete prompts will be released with the code upon publication.

ALIGNMENT JUDGE: SHARED BASE

The same base prompt powers both HALL/MIS/UNC (report-to-source) and REC (source-to-report); only the direction-specific context and rules differ.

Alignment judge (shared base)

You are given a CLAIM and a numbered list of EVIDENCE statements (1 to N). The evidence was retrieved using semantic similarity and may include irrelevant, partially relevant, or redundant items. Each evidence line is prefixed with its cosine similarity score to the claim, higher scores indicate closer semantic match but do not guarantee factual support.
{direction_context}
VERDICT DEFINITIONS.

- SUPPORTED: One or more evidence statements clearly back the claim. The core factual content of the claim is present in the evidence, even if wording, structure, or level of detail differs. Minor differences in framing are acceptable as long as the central meaning is preserved.
- PARTIALLY SUPPORTED: The evidence covers some but not all of the claim, or the match is incomplete or ambiguous (for example, confirms part of a compound statement but is silent on the rest, or provides weaker or less specific data than the claim asserts).
- NOT SUPPORTED: The evidence does not meaningfully back the claim. Use this when the claim's core assertion is absent, materially changed, or contradicted.

RULES (SHARED). (1) Use only the provided evidence text. Do not rely on outside knowledge. (2) Treat paraphrasing, summarization, and synthesis as valid support when the factual meaning matches. (3) Ignore duplicate or redundant evidence lines; they do not strengthen support. {direction_rules}
OUTPUT (STRICT JSON, NO MARKDOWN FENCES): `{"verdict": "SUPPORTED | PARTIALLY SUPPORTED | NOT SUPPORTED", "confidence": <1-5>, "justification": "<5-20 word explanation>"}`

DIRECTION: REPORT-TO-SOURCE (HALL, MIS, UNC)

Report-to-source direction

DIRECTION CONTEXT. The CLAIM is extracted from a generated report. The EVIDENCE comes from original source documents. Your task: determine whether the source evidence factually supports the report's claim. Be strict, the report must be grounded in source data.
ADDITIONAL RULES. (4) Numeric precision matters: if the claim states a specific number and the evidence gives a materially different number, that part is not supported. (5) The claim may synthesize across sources; require each key assertion to have at least one evidence match.

DIRECTION: SOURCE-TO-REPORT (REC)

Source-to-report direction

DIRECTION CONTEXT. The CLAIM is a fact extracted from an original source document. The EVIDENCE consists of passages from a generated report. Your task: determine whether the report covers this source fact. The report may paraphrase, summarise, or incorporate the fact into broader analysis, this counts as coverage.
ADDITIONAL RULES. (4) The report may express the fact at a different granularity (e.g. rounded numbers, broader category). Accept this as coverage if the core information is preserved. (5) If the report subsumes the fact into a higher-level summary that clearly encompasses it, treat it as SUPPORTED.

RELEVANCE LABELER (HIGH / RELATED / UNRELATED)

The relevance step has two implementations. The default (CLAIMWRITER experiments) is a deterministic mapping from the extraction-time relevance field (`direct` becomes HIGH, `off_topic` becomes UNRELATED, otherwise RELATED). The optional *refine* path uses the following batched LLM relevance prompt.

Batched source-fact relevance labeler

You are labeling the relevance of factual sentences to a research task. For each fact, choose exactly one label: HIGH, RELATED, or UNRELATED.
TASK: {task}
DEFINITIONS AND EXAMPLES. HIGH: directly answers the task, provides key evidence for the main question, or states a central statistic the report would need. *Example* (task: "What is Japan's elderly

population trend?"): "The proportion of people aged 65 and over in Japan reached 28.6% in 2020 and is projected to peak around 2040." (label HIGH).
RELATED: same topic or useful background but does not directly answer or strongly support the task. *Example* (same task): "The National Institute of Population and Social Security Research publishes official projections every five years." (label RELATED).
UNRELATED: does not help answer the task; off-topic or different subject. *Example* (same task): "The company reported strong earnings in the retail segment last quarter." (label UNRELATED).
FACTS TO LABEL (each line is one fact; the leading number is the fact index, 0-based): {numbered_facts}
OUTPUT. Exactly one line per fact index in the format "index: LABEL". Use only HIGH, RELATED, or UNRELATED. Example: `0: HIGH \n 1: RELATED \n 2: UNRELATED`. Output nothing else.

CLAIM AND FACT EXTRACTOR (SHARED CORE)

Source-side fact extraction and report-side claim extraction share a common system prompt (`_CLAIM_PROMPT_CORE`) of roughly four hundred lines; the report-side variant additionally requests inline citation reference IDs (`cited_refs`). We reproduce the core's skeleton; the complete prompt will be released with the code upon publication.

Claim and fact extractor (skeleton)

A claim is a self-contained, specific, information-dense statement that can stand alone in a report without any surrounding context. Include all materially useful claims; if nothing is useful for the task, return an empty list.
LANGUAGE. Output in English only, regardless of the language of the task description, report, or source content.
TASK-DRIVEN RELEVANCE. Label each claim's relevance as one of `direct` (answers, proves, or quantifies a task question), `supporting` (evidence or proof point for a direct answer), `contextual` (useful background), or `off_topic` (no meaningful connection). Score salience 0-10 (10 = answers a primary task question with hard data; 0 paired with off_topic).
SELF-SUFFICIENCY. Each claim must include `text` (fully standalone, no pronouns or generic placeholders, named subject inline), `time_reference` (concrete calendar anchor only, never placeholder prose), `entities` (specific proper-noun named entities, no generic categories), `relevance`, and `salience_score`.
HARD RULES. No vague references ("it", "the company", "the study"); no relative time unless converted to a concrete year/date; no meta placeholders in `time_reference` (e.g. "report publication year" without digits); no bare metrics without a named subject; no claim that depends on its heading or neighbours for meaning; preserve numeric precision; exclude boilerplate, navigation, and legal/privacy text.
QUALITY GATE. Apply five checks to every claim before output: subject identifiable, event identifiable, time anchored (or correctly empty), scope present when available, claim serves a task section. If any fails, rewrite or drop.
OUTPUT (JSON). `{"claims": [{"text", "time_reference", "entities", "relevance", "salience_score"}]}`. Report-side adds `cited_refs`: list of integer reference IDs found in or near the claim text.

FACT-DIFF (STAGE 1) SOURCE-CHANGE CLASSIFIER

This prompt is part of the writer's incremental-update path (Section 3). For each changed source, the model receives the original page, the updated page, and the facts previously extracted from that source. It classifies substantive differences as replacements, removals, or additions. The exact prompt will be released with the writer code upon publication; the schema used in our experiments is summarized below.

Source-change classifier (schema)

INPUT. The user task, source title and URL, original page text, updated page text, and the previously extracted facts from that source as `[{id: fN, text: ...}, ...]`.
TASK. Place each substantive difference into exactly one category. `replaced` contains an existing fact whose value changed and returns its identifier, corrected self-contained text, and a short change description. `removed_ids` contains facts no longer supported by the updated page and for which no replacement value exists. `added` contains genuinely new facts that were not covered by the previous fact set. Pure wording, formatting, navigation, and boilerplate changes are ignored; replacement takes priority when an old fact receives a new value.

OUTPUT (JSON). {"replaced": [{"id", "new_text", "change"}], "removed_ids": ["fN"], "added": [{"text", "excerpt", "relevance_score", "specificity_score"}]}.