Beyond Single-shot Writing: Deep Research Agents are Unreliable at Multi-turn Report Revision

Bingsen Chen[1,2,*], Boyan Li[3,†,*], Ping Nie[5], Yuyu Zhang[6], Xi Ye[3,4,‡], Chen Zhao[1,2,‡],
[1]New York University [2]NYU Shanghai [3]University of Alberta
[4]Princeton University [5]University of Waterloo [6]Verdent AI, Inc.

Abstract

Existing benchmarks for Deep Research Agents (DRAs) treat report generation as a single-shot writing task, which fundamentally diverges from how human researchers iteratively draft and revise reports via self-reflection or peer feedback. Whether DRAs can reliably revise reports with user feedback remains unexplored. We introduce MR DRE, an evaluation suite that establishes multi-turn report revision as a new evaluation axis for DRAs. MR DRE consists of (1) a unified long-form report evaluation protocol spanning comprehensiveness, factuality, and presentation, and (2) a human-verified feedback simulation pipeline for multi-turn revision. Our analysis of five diverse DRAs reveals a critical limitation: while agents can address most user feedback, they regress on 16–27% of previously covered content and citation quality. Over multiple revision turns, even the best-performing agent leaves significant headroom, as they continue to disrupt content outside the feedback's scope and fail to preserve earlier edits. We also show that these issues are not easily resolvable through inference-time fixes such as prompt engineering and a dedicated sub-agent for revision[1].

1 Introduction

Recent advances in the agentic capabilities of language models have led to the rise of Deep Research Agents (DRAs) (OpenAI, 2025; Perplexity, 2025; Tongyi, 2025; Shao et al., 2025). DRAs tackle complex research queries by extensively searching and browsing the web, then synthesizing large volumes of information into long-form reports with rich citations and well-organized structure.

However, what dimensions to evaluate these complex systems against remains an open problem. Early attempts (Li et al., 2025b; Tongyi, 2025) used multi-hop QA benchmarks (Wei et al., 2025a; Chen et al., 2025; Phan et al., 2025; Mialon et al., 2023) to evaluate DRAs' multi-step retrieval and reasoning ability. However, such benchmarks rely on short-form answers and fail to capture the report-writing capabilities of DRAs. More recent work has begun to evaluate long-form report generation directly (Du et al., 2025; Yao et al., 2025; Sharma et al., 2025; Xu et al., 2025), but uniformly treat it as a *single-shot* task: given a query, agents gather information and produce a report in one pass. This departs from human practice, where reports are produced through iterative revision, often guided by self-reflection or feedback.

In this work, we propose **multi-turn report revision** as a new evaluation axis for DRAs. In this setting, DRAs revise an initial report over multiple

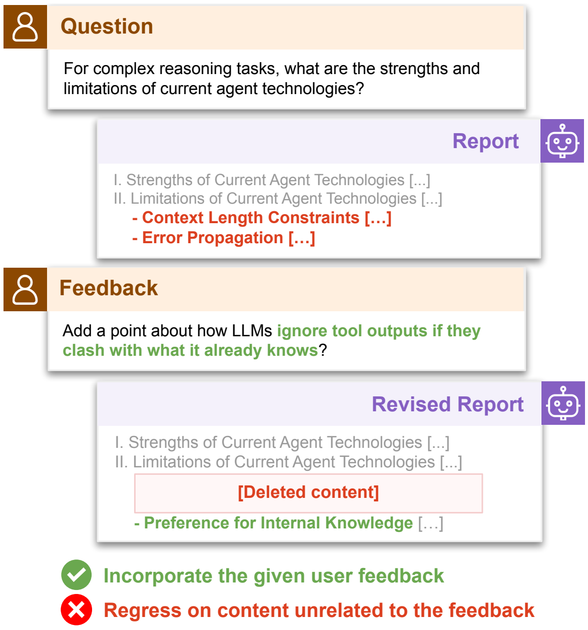

Figure 1: **Illustrative example of multi-turn revision failure in Deep Research Agents.** The revised report incorporates the user feedback but removes previously covered content that is outside the feedback's scope.

[*]Equal contribution. [‡]Equal advising. [†]Work done when visiting NYU Shanghai.

[1]Our code and data are released at: `https://github.com/BaleChen/Mr-Dre`.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 13325–13356

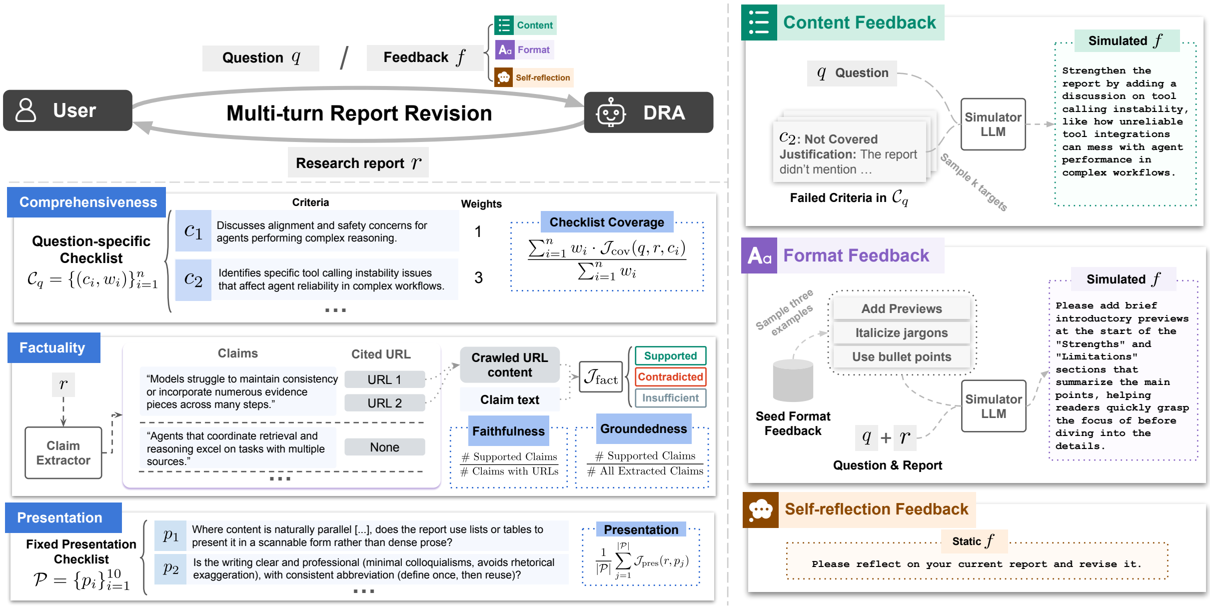

Figure 2: **MR DRE Evaluation Suite.** In multi-turn report revision, a DRA iteratively drafts and revises a report r for question q given user feedback f (top left). MR DRE provides a unified Deep Research report evaluation protocol (bottom left) along three dimensions: Comprehensiveness, Factuality, and Presentation. To evaluate multi-turn revision performance, MR DRE provides a pipeline to simulate content, format, and self-reflection feedback (right).

turns in response to user feedback. This capability is important for two reasons. First, in practical deployment, users do not often accept an initial response as-is (Lee et al., 2025a), and may request additional details or changes in structure. Agents that cannot effectively incorporate such feedback while preserving the quality of the rest of the report provide limited utility. Second, iterative revision offers a natural mechanism for improving report quality with additional compute. While test-time scaling has proven effective for reasoning (Snell et al., 2025; Muennighoff et al., 2025), its impact on report generation remains unexplored. We therefore ask: **Can current deep research agents reliably improve their reports via user feedback?**

To study this, we introduce **M**ulti-turn **R**evision of **D**eep **R**esearch Agent **E**valuation (**MR DRE**), a new evaluation suite supporting iterative report writing. For report evaluation, MR DRE unifies evaluation practices from prior benchmarks, which have adopted divergent metrics, into a lean protocol covering three dimensions: comprehensiveness, factuality, and presentation (Figure 2, left). To enable multi-turn report revision evaluation, MR DRE provides a human-verified feedback simulation pipeline that generates realistic user feedback on report content and formatting (Figure 2, right).

With MR DRE, we evaluate five diverse DRAs and find that current systems cannot reliably revise reports in response to user feedback. Although agents address over 90% of requested edits, content or format revisions frequently reduce overall report quality: 16–27% of previously covered content is broken, and citation quality degrades. Moreover, when extending revision to multiple turns, some agents show minimal to negative progress, and even the best-performing agents leave significant gaps as they continue to disrupt content outside feedback's scope and fail to preserve edits made in earlier turns. Finally, these failures persist despite inference-time remedies, including extensive prompt engineering and a dedicated reviser sub-agent. These findings suggest that reliable multi-turn revision will require more fundamental changes in DRA training and scaffold design, which we advocate as a priority for future research.

In summary, our main contributions are:

- **A novel evaluation axis for DRAs.** We identify multi-turn report revision as an important yet underexplored capability of DRAs.
- **MR DRE evaluation suite.** MR DRE unifies prior evaluation practices into a concise three-dimensional protocol for Deep Research report generation and includes a human-verified feedback simulation pipeline for multi-turn revision.
- **Comprehensive analysis of current DRAs' multi-turn revision ability.** We reveal systematic limitations in current DRAs' ability to revise reports from user feedback and show that inference-time fixes are insufficient.

2 Task Definition

We begin by formalizing deep research and defining our multi-turn report revision task: Unlike standard search-augmented language models that are tasked with generating short-form (Yang et al., 2018; Wei et al., 2025a) or paragraph-long answers (Fan et al., 2019; Han et al., 2024), a Deep Research Agent is a system of one or multiple LLMs augmented with web searching tools that autonomously retrieves and analyzes vast online information and synthesizes findings into a comprehensive, well-cited research report.

Formally, given a user's initial query q, a DRA $\mathcal{A}$ generates a report $r_1 = \mathcal{A}(q)$. Current deep research benchmarks evaluate only this single output r_1, treating report writing as a one-shot task. We extend this paradigm to *multi-turn report revision*. After receiving the initial report r_1, a user may provide feedback f_1, prompting the agent to revise the report and yield $r_2 = \mathcal{A}(q, r_1, f_1)$. This process can continue iteratively: at turn t, the agent produces $r_t = \mathcal{A}(q, r_1, f_1, \ldots, r_{t-1}, f_{t-1})$, conditioning on all previous turns of drafts and feedback.

3 A Unified Deep Research Report Evaluation Protocol

In this section, we introduce the first component of Mr Dre, a comprehensive protocol for Deep Research report evaluation. We start with introducing our curated data (§3.1), and then detail the evaluation dimensions (§3.2).

3.1 Data Curation

To enable reliable evaluation, we guide our data curation with two criteria. First, each question must be paired with a *question-specific checklist*: a set of content criteria that a high-quality report should satisfy. Such checklists provide ground-truth coverage targets tailored to each question and have proven effective for evaluating complex, long-form generations (Ruan et al., 2025; Lee et al., 2025b). Second, the questions and checklists must be sufficiently complex that DRAs are unlikely to cover all checklist items on their first attempt, leaving meaningful room for iterative revision.

Following these criteria, we include expert-annotated research questions from three datasets: ResearchRubrics (Sharma et al., 2025), RigorousBench (Yao et al., 2025), and ResearcherBench (Xu et al., 2025). These datasets feature expert-annotated research-intensive questions paired with evaluation checklists for improvement. Table 4 summarizes the dataset statistics.

	Checklist Comp.	Claim Fact.	Report Pres.	Ref.-free	Multi-turn
ResearchRubrics	✓	✗	✗	✓	✗
RACE&FACT	✓	✓	✓	✗	✗
RigorousBench	✓	✗	✓	✓	✗
ResearcherBench	✓	✗	✗	✓	✗
DR-ReportEval	✗	✓	✓	✗	✗
DeepEval	✓	✓	✓	✓	✗
Mr Dre	✓	✓	✓	✓	✓

Table 1: **Comparison of Mr Dre and previous Deep Research report evaluation protocols.** Mr Dre checks all five aspects: checklist-based comprehensiveness evaluation, claim-level factuality assessment, report presentation scoring, requiring no reference answer, and support for multi-turn report revision (§4).

3.2 Evaluation Dimensions & Pipeline

We combine best practices from prior benchmarks into three dimensions: *comprehensiveness*, *factuality*, and *presentation*. These dimensions capture a minimal set of qualities that characterize an excellent research report: it should cover all essential information, make accurate citations and well-supported claims, and present content in a clear, well-organized manner. We illustrate our evaluation protocol in Figure 2 (left), and compare Mr Dre's unified protocol with previous benchmarks' evaluation in Table 1. Note that our protocol is generally applicable to all Deep Research report generation tasks and can easily integrate new datasets.

Comprehensiveness A high-quality research report should cover all relevant aspects of a research question. We measure this using the coverage score of question-specific checklists. A question q's checklist $\mathcal{C}_q = \{(c_i, w_i)\}_{i=1}^n$ consists of n criteria, where each criterion c_i has a weight w_i reflecting its importance. Following prior work (Sharma et al., 2025; Yao et al., 2025), we adopt ternary grading to allow partial credit: an LLM judge $\mathcal{J}_{\text{cov}}$ evaluates each criterion against the question q and report r, assigning a score $s_i = \mathcal{J}_{\text{cov}}(q, r, c_i) \in \{0, 0.5, 1\}$ corresponding to absent, partial, or complete coverage. The coverage score is the weighted average:

$$\textsc{Cov}(r) = \frac{\sum_{i=1}^n w_i \cdot s_i}{\sum_{i=1}^n w_i}.$$

Factuality A high-quality research report should make accurate claims backed by reliable citations. Based on the in-line citations, we evaluate factuality from two angles: *citation faithfulness* that measures the proportion of cited claims that are actually supported by their referenced sources, and *claim groundedness* that measures the proportion of all claims that can be verified against external evidence.

Specifically, we adapt VeriScore (Song et al., 2024) for Deep Research report evaluation. Given a report r, we first extract the set of atomic claims $\mathcal{E}(r) = \{e_1, e_2, \dots, e_m\}$ using an LLM, where each claim e_i is associated with zero, one, or multiple cited URLs. An LLM judge $\mathcal{J}_{\text{fact}}$ then evaluates each claim against the crawled content of its cited sources, classifying it as supported, contradicted, or insufficiently evidenced. We define citation faithfulness (left) and claim groundedness (right) as:

$$\textsc{Fa}(r) = \frac{|\mathcal{S}|}{|\mathcal{E}_{\text{cited}}(r)|}, \qquad \textsc{Gr}(r) = \frac{|\mathcal{S}|}{|\mathcal{E}(r)|}$$

where $\mathcal{E}_{\text{cited}}(r) \subseteq \mathcal{E}(r)$ is the subset of claims with at least one citation, and $\mathcal{S} \subseteq \mathcal{E}(r)$ denotes the subset of supported claims. We provide additional technical details in Appendix B.2.

Presentation A high-quality research report should organize dense information into a readable, well-structured format with professional language. We consolidated and refined prior works' (Fan et al., 2025; Yao et al., 2025; Wang et al., 2025) divergent criteria to arrive at a unified checklist of 10 questions (listed in Table 5). For each criterion p_j in the checklist $\mathcal{P} = \{p_1, \dots, p_{10}\}$, an LLM judge $\mathcal{J}_{\text{pres}}$ assigns a binary score. The overall presentation score is calculated as:

$$\textsc{Pre}(r) = \frac{1}{|\mathcal{P}|} \sum_{j=1}^{|\mathcal{P}|} \mathcal{J}_{\text{pres}}(r, p_j)$$

4 Extending Evaluation to Multi-turn Report Revision

MR DRE also introduces an automated yet human-validated pipeline for multi-turn report revision evaluation. In this section, we introduce (1) a reliable way to simulate realistic user feedback (illustrated in Figure 2, right), and (2) metrics to measure revision success.

4.1 Feedback Simulation

We design a feedback simulation pipeline that generates diverse and realistic user feedback on Deep Research reports. We consider three feedback categories corresponding to distinct revision settings: <u>content</u>, <u>format</u>, and <u>self-reflection</u>.

Content Feedback requests adding new information or correcting existing content. Under this setting, a successful revision must address the feedback targets while preserving unrelated report content. To simulate such feedback, we leverage the checklist-based evaluation from §3.2. Given a report draft, we first evaluate all checklist criteria using $\mathcal{J}_{\text{cov}}$, which we also ask for a justification for each score. We then uniformly sample k uncovered criteria as *feedback targets*, denoted $\mathcal{T}^{(t)} \subset \mathcal{C}_q$ for turn t, and prompt a feedback simulator LLM to generate natural feedback based on the question, the sampled feedback targets, and their corresponding scores and justifications. Grounding feedback in scoring justification rather than raw criteria text produces more natural requests that reflect how users would articulate what is missing.

Format Feedback targets the report's structure, style, or presentation. We expect DRAs to incorporate the formatting feedback without disrupting existing content coverage. We curate 21 seed format feedback examples covering common user requests (e.g., adding bullet points, improving sectioning, or including TL;DR summaries. Full list in Table 7). Given a report draft, we randomly sample three seed examples and prompt the LLM to select the most applicable one, then expand it into a piece of feedback specific to the report draft. Using seed examples guides the simulator toward generating realistic feedback, while random sampling ensures diversity.

Self-Reflection Feedback provides no explicit revision guidance. The feedback is simply: "`Please reflect on your current report and revise it.`" This setting tests whether DRAs can autonomously identify and address deficiencies.

Human Validation. We conduct human annotation to validate simulated content and format feedback along four dimensions: *naturalness*, *draft-specificity*, *actionability*, and, for format feedback, *content preservation*. Our pipeline achieves near-ceiling scores across all dimensions with high inter-annotator agreement. Details are in Appendix C.3.

4.2 Measuring the Success of Revision

We introduce two additional metrics to measure the effectiveness of revision via the question-specific checklists. Let $s_i^{(t)}$ denote the coverage score of criterion c_i for report $r^{(t)}$.

			ResearchRubrics				RigorousBench				ResearcherBench				Avg.	Avg.
DRA	**Turn**	**Type**	**Cov.**	**Fa.**	**Gr.**	**Pre.**	**Cov.**	**Fa.**	**Gr.**	**Pre.**	**Cov.**	**Fa.**	**Gr.**	**Pre.**	**Inc.**	**Brk.**
OpenAI DR	1	–	62.3	63.6	28.8	97.9	42.2	63.5	33.5	97.2	68.5	80.3	41.4	95.6	–	–
	2	Reflect	-1.5	-10.3	-8.4	+1.0	-0.8	-2.4	-4.4	+0.7	-3.9	-7.0	-5.3	+2.9	–	14.9
		Content_1	-6.6	-10.3	-6.1	-5.1	-2.0	-16.9	-11.7	-5.4	-9.1	-16.4	-11.0	-4.6	93.6	29.7
		Format	-2.2	-19.9	-13.0	+0.2	-3.6	-8.8	-8.4	0.0	-3.9	-14.4	-7.1	+1.4	98.5	14.8
Sonar DR	1	–	70.9	71.7	56.4	90.8	55.2	76.4	64.1	90.3	80.0	75.3	64.8	89.6	–	–
	2	Reflect	-5.2	-55.6	-48.1	+4.6	-7.3	-60.8	-54.0	+2.5	-5.2	-67.4	-59.1	+6.0	–	20.5
		Content_1	-11.5	-26.7	-19.7	+0.2	-8.3	-23.3	-21.9	-1.1	-10.4	-8.5	-3.9	+0.6	92.6	34.3
		Format	-16.2	-35.6	-28.8	-6.1	-13.9	-35.9	-32.6	-9.4	-13.4	-30.0	-27.1	-3.1	85.5	27.8
LC ODR	1	–	60.9	72.5	39.7	99.5	43.0	74.2	46.1	99.8	71.0	82.6	51.8	99.8	–	–
	2	Reflect	+3.2	-4.8	-4.5	0.0	+3.8	-3.3	-1.0	0.0	+3.8	-8.5	-10.4	-0.2	–	8.5
		Content_1	-11.5	-6.5	-8.4	-1.3	-5.3	-5.2	-7.1	-1.9	-8.7	-9.1	-19.4	-0.5	93.0	38.6
		Format	-5.8	-1.4	-3.8	-1.9	-3.5	-5.1	-8.6	-2.1	-9.9	-4.2	-14.4	-2.4	91.1	23.9
Tongyi DR	1	–	58.5	–	–	99.3	39.4	–	–	99.2	68.4	–	–	99.9	–	–
	2	Reflect	-0.3	–	–	-1.2	+0.2	–	–	+0.1	-0.7	–	–	+0.2	–	9.9
		Content_1	-9.2	–	–	-5.3	-0.9	–	–	-1.2	-5.0	–	–	-0.8	90.2	31.5
		Format	-6.7	–	–	-2.6	-5.8	–	–	-2.5	-9.8	–	–	-4.1	94.3	23.2
DR Tulu	1	–	60.7	64.7	46.3	98.2	42.8	63.4	49.2	96.7	67.5	79.0	62.1	98.5	–	–
	2	Reflect	-0.7	-3.9	-0.4	-1.1	-0.7	-2.6	+0.3	-1.9	-0.1	-5.7	-3.5	-1.1	–	11.4
		Content_1	-2.7	-5.8	-2.6	-3.5	+2.5	-4.0	-1.7	-1.1	+1.0	-4.6	-6.8	0.0	90.3	23.5
		Format	-1.0	-2.2	+2.2	-1.3	-2.2	+1.0	+2.7	-1.7	-3.1	-5.7	-2.6	+0.7	94.9	14.0

Table 2: **Main Results.** We report the Coverage (Cov.), Citation Faithfulness (Fa.), Claim Groundedness (Gr.), and Presentation (Pre.) score in percentage points. Incorporation (Inc.) and Break (Brk.) rate results are averaged across three datasets. For the second turn, we show the score changes from the first turn results for all evaluation metrics and feedback types, where improvement is colored in green and reduction is colored in red.

Incorporation Rate measures whether the revision successfully incorporates feedback at turn t:

$$\text{INC} = \begin{cases} \frac{1}{|\mathcal{T}^{(t)}|}\sum_{c_i \in \mathcal{T}^{(t)}} \mathbb{1}\left[s_i^{(t)} = 1\right], & \text{Content feedback} \\ \mathcal{J}_{\text{inc}}(f_t, r_{t-1}, r_t), & \text{Format feedback} \end{cases}$$

For content feedback, this is the proportion of feedback targets $\mathcal{T}^{(t)}$ that achieve full coverage after revision. For format feedback, an LLM judge $\mathcal{J}_{\text{inc}}$ assesses whether the formatting suggestion is followed (binary scoring).

Break Rate is the proportion of previously covered criteria whose scores degrade after revision:

$$\text{BRK} = \frac{|\{c_i : s_i^{(t-1)} > 0 \wedge s_i^{(t)} < s_i^{(t-1)}\}|}{|\{c_i : s_i^{(t-1)} > 0\}|}$$

A break rate of 0 indicates that the revision preserves all previously covered content, while a high break rate suggests making destructive edits that fix one issue at the cost of disrupting others.

All metrics are defined per sample and reported as an average across the dataset. Prompt templates are in Appendix F.

5 Experiments

Using MR DRE, we study our central research question: **how reliably can current DRAs revise reports via user feedback?** We first examine various DRAs' revision performance under three distinct feedback settings, and then analyze how revision behavior changes as we scale the number of turns (§5.2) and feedback targets (§5.3).

Settings. We evaluate five DRAs under three revision settings: self-reflection (Reflect), content feedback (Content_k), and format feedback (Format). In the Content_k setting, each content feedback message targets k checklist criteria that the previous-turn draft fails to satisfy. Due to the high cost of running proprietary DRAs, we conduct only the second-turn experiments on the three complete datasets in Table 4. For experiments with up to 4 turns and varying the number of feedback targets (k), we use a sub-sampled Core Set (25 questions from each dataset, totaling 75 questions). Details about the subset creation are in Appendix A.2. Given that the size of Deep Research benchmarks are generally small, we provide statistical significant test results of our main observations in Appendix E.5.

Evaluated Agents. We evaluate five DRA systems under three categories: (1) **Proprietary scaffold and model(s):** OpenAI o4-mini Deep Research (OpenAI DR) (OpenAI, 2025) and Sonar Deep Research by Perplexity (Sonar DR) (Perplexity, 2025). Such systems reveal little information about the agents' details. (2) **Open scaffold, pro-**

prietary models: LangChain Open Deep Research (LC ODR) (LangChain, 2025). It orchestrates a system of research, summarization, and finding compression agents, and a report-writing agent. We used GPT-4.1-Nano for summarization and GPT-4.1-mini for the rest. (3) **Open scaffold and models:** Tongyi Deep Research (Tongyi DR) (Tongyi, 2025) and DR Tulu (Shao et al., 2025), which are post-trained for Deep Research report generation. Tongyi DR is not trained to generate citations, so we omit its citation-related results.

5.1 Main Results

Table 2 shows the first and second turn results under three feedback settings. We observe that:

DRAs struggle to reliably improve, or even preserve, report comprehensiveness across almost all feedback settings. Across different DRAs, feedback types, and datasets, coverage scores predominantly decrease from Turn 1 to Turn 2. Under self-reflection, only LC ODR achieves a modest coverage gain (+3.6%), while all other DRAs exhibit decreases or negligible changes. Even in the Content_1 setting, where the feedback explicitly identifies an unsatisfied checklist criterion, all agents except DR Tulu suffer coverage drops ranging from -2.0% to -11.5%. Format feedback, which by design should preserve content, leads to universal coverage drop across DRAs (-1.0% to -16.2%). Sonar DR, although achieving the best performance in the initial turn, also shows the largest performance drop across all feedback settings. These patterns indicate a systematic limitation in current DRAs' ability to revise reports based on different types of user feedback.

While DRAs can follow most of the feedback instructions, they fail to preserve content outside the feedback's scope. To understand why coverage degrades after revision, we examine the incorporation and break rates. All DRAs demonstrate strong instruction-following capabilities: incorporation rates mostly exceed 90% for both content and format feedback. However, this success comes at the cost of disrupting previously satisfied content. Break rates average 31% under content feedback and 21% under format feedback, as a substantial fraction of earlier coverage is lost after revision. Interestingly, break rates are lower under self-reflection, suggesting that more targeted feedback induces more aggressive edits that inadvertently affect unrelated content.

Revision significantly degrades citation faithfulness and claim groundedness. Beyond content coverage, factuality metrics also deteriorate after revision. The underlying causes vary across agents, such as fewer supported claims, and we provide a detailed analysis in Appendix E.1. Notably, Sonar DR exhibits the most severe degradation, especially after self-reflection, with faithfulness plummeting by up to -67.4% and groundedness by up to -59.1%. Further inspection reveals that Sonar DR produces reports with no citations 68% of the time after receiving self-reflection feedback.

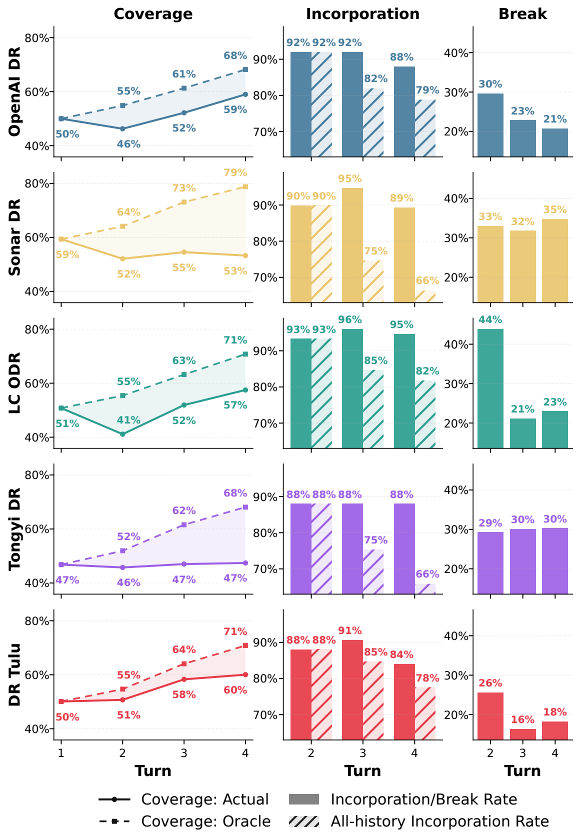

Figure 3: **Results for extending to 4 turns of revision under Content_1 setting**. We report the (left) checklist coverage (actual vs. oracle), (middle) incorporation rate, and (right) break rate.

5.2 How well can DRAs revise reports when extending to multiple turns?

We extend revision up to four turns under the Content_1 and Reflect setting. We found that:

DRAs fail to effectively accumulate coverage gains over multiple turns of content feedback. On the left of Figure 3, we show each agent's coverage score (solid line) alongside the oracle score (dashed line), which represents the upper bound performance assuming perfect incorporation and zero break rate from Turn 1 onward. Tongyi DR and Sonar DR show minimal or even negative progress over multiple turns, while others achieve gradual coverage improvements with additional content feedback rounds. However, all agents lag far behind the oracle, and this gap shows no sign of closing over turns: by Turn 4, the gap between actual and oracle scores ranges from 9% (OpenAI DR) to 26% (Sonar DR).

As shown on the right of Figure 3, the persistent gap from the oracle traces to both imperfect incorporation rates and non-trivial break rates across all agents. Notably, all DRAs break previously satisfied content at around 20-30% by Turn 4. Tongyi DR and Sonar DR have consistently high break rates that offset any gains from incorporating content feedback over turns, whereas OpenAI DR, DR Tulu, and LC ODR show decreasing break rates (from 33% to 21% on average).

DRAs break not only content outside of feedback's scope but also feedback targets from previous turns. To measure this, we report the all-history incorporation rate: the proportion of feedback targets from all previous turns that remain satisfied at turn t. While current-turn incorporation stays stable around 90%, all-history incorporation drops substantially. For instance, Sonar DR's all-history incorporation rate falls from 90% at Turn 2 to 66% by Turn 4. This gap indicates that agents fail to preserve previous fixes while addressing new feedback, even though earlier feedback remains in the input context.

We present additional multi-turn results in Appendix E.2. We found that citation faithfulness and claim groundedness decrease over turns across all DRAs under the Content_1 setting. Also, multiple turns of self-reflection similarly degrade coverage, citation faithfulness, and claim groundedness. These findings further underscore the unreliability of current DRAs in multi-turn report revision.

5.3 How reliable are DRAs given feedback with multiple targets?

We then investigate how content feedback targeting multiple checklist criteria affects revision performance, which we show in Figure 4.

Increasing the number of feedback targets leads to higher coverage gains across all DRAs. All DRAs consistently achieve higher coverage as more unsatisfied criteria are given in the feedback. To understand this pattern, we examine the incorporation and break rates as k increases: Incorporation

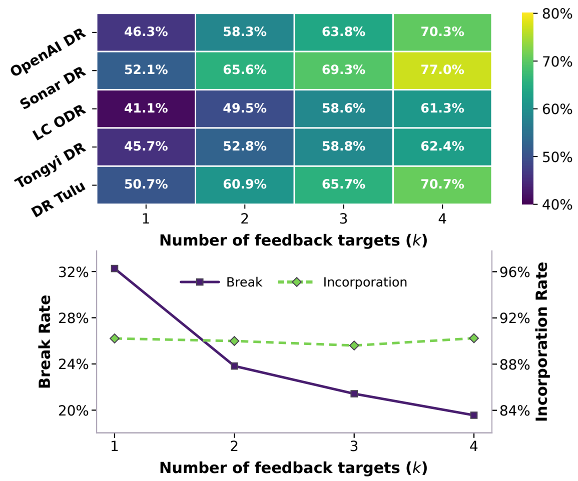

Figure 4: **Coverage (Top), Break rate, and Incorporation rate (Bottom) with varying k.** Break and incorporation rates are averaged across 5 DRAs since they all show the same trend.

rates remain high regardless of k, as agents can effectively handle multiple feedback targets at once. Interestingly, break rates also decrease with larger k, suggesting that agents make less disruptive edits when given more targets to fix.

6 Can Inference-time Fixes Improve Revision Performance?

Our analysis reveals that DRAs cannot reliably revise reports based on user feedback, due to disruptive edits on existing content and citations, compounded by imperfect incorporation of the feedback. In this section, we investigate whether simple inference-time fixes can address these limitations without heavily modifying the underlying DRA system. Specifically, we test two approaches:

Prompt Engineering (PE) converts user feedback into a structured edit plan before revision. This approach decomposes the feedback into concrete, step-by-step instructions using an LLM (see example in Figure D.1), with the hypothesis that explicit guidance may help agents make more targeted edits without affecting unrelated content.

Reviser Sub-agent (Reviser) delegates the revision task to an LLM augmented with a web search tool. Since DRAs are optimized for multi-step tool calling and reasoning rather than localized editing, we hypothesize that a separate LLM with strong instruction-following capabilities can better incorporate user feedback while preserving content outside the feedback's scope.

DRA	Setting	Core Set Cov.	Fa.	Gr.	Pre.	Inc.	Brk.
OpenAI DR	Initial	50.0	74.6	37.7	97.0	–	–
	$Content_1$	-3.7	**-22.2**	-13.9	-3.5	91.9	29.6
	+PE	+2.2	-26.3	-13.8	-1.8	93.2	16.1
	+Reviser	**+5.1**	-30.4	**-9.4**	**+1.0**	**94.6**	**10.7**
	Format	-4.5	-19.9	-12.6	**-0.3**	97.3	19.1
	+PE	-3.7	-31.7	-17.3	**-0.3**	**100.0**	**14.5**
	+Reviser	**-3.1**	**-16.3**	**-11.9**	-0.7	98.7	16.8
DR Tulu	Initial	50.1	68.7	53.2	97.7	–	–
	$Content_1$	+0.6	-4.1	-4.0	-2.3	88.0	25.6
	+PE	+3.4	**+1.3**	**+0.9**	-3.4	88.0	14.1
	+Reviser	**+5.8**	-8.4	-11.3	**-2.1**	**92.0**	**9.5**
	Format	-0.9	-1.8	**+1.5**	**-0.9**	94.0	13.1
	+PE	-0.8	**-0.6**	+0.5	-1.5	96.0	**12.1**
	+Reviser	**-0.3**	-1.4	-20.1	-1.3	**100.0**	12.6

Table 3: **Results of PE and Reviser on the Core Set.** Second turn's score change from Initial (turn 1) is shown for the four main metrics, along with incorporation (Inc.) and break (Brk.) rates. Best values are bolded within each agent and setting.

Experimental Setup We evaluate the two fixes on OpenAI DR and DR Tulu under the $Content_1$ and Format settings using the Core Set. For PE, we use GPT-4.1 to transform simulated feedback into structured edit plans before sending them to the DRA. For the Reviser, we implement a ReAct agent (Yao et al., 2023) with Qwen3-30B-A3B-Instruct, a model with strong instruction-following capabilities (84.7% on IFEval (Zhou et al., 2023)), augmented with Google Search to retrieve additional information when needed. We include details and prompts in Appendix D.

6.1 Findings

Both approaches can improve coverage by reducing break rates and improving incorporation rates. As shown in Table 3, both PE and the Reviser enable DRAs to achieve coverage improvements for both content and format feedback, compared to when no fix is applied. These improvements stem from consistently higher incorporation rates and lower break rates. The Reviser generally outperforms PE in coverage scores, likely because the model is optimized for instruction-following and thus can execute edit requests more faithfully without introducing disruptive changes.

However, these fixes fall short of fully addressing the challenges in multi-turn revision. First, even with the best-performing Reviser, agents still break over 10% of previously covered criteria on average, especially under the format feedback setting, where both fixes yield smaller gains. Second, neither approach resolves citation degradation: for OpenAI DR, both PE and the Reviser still show substantial drops in citation faithfulness and claim groundedness compared to the initial report. These persistent gaps indicate that inference-time mitigations alone cannot fully address the multi-turn revision challenges. Achieving reliable report revision that preserves both content coverage and citation quality will likely require more fundamental advances in training algorithms or scaffold design.

7 Related Works

Deep Research Report Evaluation The emergence of DRAs has motivated long-form report benchmarking with varied evaluation approaches. Some rely on gold-standard reference reports to judge comprehensiveness (Du et al., 2025; Li et al., 2025a), while others adopt checklist-based evaluation (Hashemi et al., 2024; Lee et al., 2025b; Arora et al., 2025) specifies question-specific criteria to measure content coverage (Wang et al., 2025; Yao et al., 2025; Xu et al., 2025; Sharma et al., 2025). A complementary axis concerns factual verifiability. Prior works assessed it via citation quality (Gao et al., 2023; Ye et al., 2024; Liu et al., 2023), which is widely adopted in recent Deep Research evaluations (Fan et al., 2025; Yao et al., 2025; Xu et al., 2025; Li et al., 2025a). Our MR DRE builds upon these evaluation practices to arrive at a unified protocol, meanwhile extending the scope to multi-turn report revision, an ability that remains underdeveloped for current DRAs.

Revision Abilities of LLMs Prior works have found that LLMs can improve their reasoning, coding, and agentic task performance through self-reflection (Madaan et al., 2023; Zelikman et al., 2024; Shinn et al., 2023). Yet, similar to our findings, some have also shown contradictory results that such gains can be fragile, as LLMs often fail to identify their own mistakes and thus struggle to self-correct reliably (Huang et al., 2023; Lee et al., 2025a). Also, another line of work obtains feedback from external tools or critic models for LLMs to revise their outputs (Gou et al., 2023; Nathani et al., 2023; Jiang et al., 2023; Wadhwa et al., 2024). In this work, we extend the discussion to Deep Research multi-turn report revision, examining both self-reflection and user feedback settings. Although Qiao et al. (2025) and Han et al. (2025) explored iterative drafting for DRAs, they did not consider multi-turn user feedback settings, where we reveal critical limitations and provide a comprehensive testbed for future development.

8 Conclusions

In this paper, we propose multi-turn report revision as an essential yet overlooked capability of DRAs. We introduce MR DRE, an evaluation suite featuring a unified evaluation protocol for long-form reports and a human-verified feedback generation pipeline for simulating user feedback in multi-turn revision. Across five diverse DRAs and three feedback settings, current systems cannot reliably improve reports through revision. While DRAs mostly address the given feedback, they frequently regress on unrelated content and citation quality. These gaps are not easily closed by simple fixes such as prompt engineering or dedicated sub-agents. We view multi-turn report revision as a critical missing piece in developing useful DRAs, and MR DRE aims to drive progress toward agents that can both conduct complex research tasks and reliably adapt to users' evolving needs.

Limitations

Understanding the Causes of Unreliability While our work reveals critical limitations in DRAs' multi-turn revision ability, the causes of the high break rate, imperfect incorporation rate, and citation degradation are not yet fully understood (see error cases in Appendix G.2). We encourage future work to systematically analyze these underlying causes, which would inform the development of new training algorithms or agent scaffolds for reliable multi-turn revision.

Model Scaling Effects on Revision Ability. Due to the high cost of running proprietary models on Deep Research tasks, we did not investigate how scaling up the backbone model affects revision reliability. For instance, we used o4-mini-deep-research instead of the stronger o3-deep-research for OpenAI DR, and LC ODR uses GPT-4.1-mini as its backbone. Although we provide a small-scale experiment with o3-deep-research in Appendix E.4, how model scaling affects revision ability remains unclear and warrants more comprehensive investigation.

Missing Considerations in Evaluation Protocol. First, our feedback simulation assumes that the questions and checklists are high-quality. Future work could enhance the feedback simulation pipeline to be robust to varying checklist quality, potentially incorporating LLM-based checklist evaluation (Lee et al., 2025b; Wei et al., 2025b). Second, real user feedback could be ambiguous and even contradictory. Such complexity is not yet considered in MR DRE feedback simulation. Our results already demonstrate that current DRAs frequently fail even when given concrete, actionable feedback; we would expect such complex feedback to pose an even greater challenge, but an interesting future direction to work on. Lastly, MR DRE does not penalize excessive report length. We observe that Sonar DR consistently achieves higher coverage, partially because its reports are substantially longer than those of other DRAs (on average 9452 tokens for Sonar DR vs 4516 tokens for other DRAs per report), though it also has a lower presentation score due to consistently failing p_3 in Table 5. However, ideal length varies across questions and user preferences, making it difficult to define an evaluation scheme. Future work could build upon the MR DRE protocol to enhance length-aware evaluation.

Acknowledgements

This work was supported in part through the NYU IT High Performance Computing resources, services, and staff expertise.

References

Rahul K Arora, Jason Wei, Rebecca Soskin Hicks, Preston Bowman, Joaquin Quiñonero-Candela, Foivos Tsimpourlas, Michael Sharman, Meghan Shah, Andrea Vallone, Alex Beutel, and 1 others. 2025. Healthbench: Evaluating large language models towards improved human health. *arXiv preprint arXiv:2505.08775*.

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. 2025. Browsecomp-plus: A more fair and transparent evaluation benchmark of deep-research agent. *arXiv preprint arXiv:2508.06600*.

Mingxuan Du, Benfeng Xu, Chiwei Zhu, Xiaorui Wang, and Zhendong Mao. 2025. Deepresearch bench: A comprehensive benchmark for deep research agents. *arXiv preprint arXiv:2506.11763*.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. ELI5: Long form question answering. In *Proceedings of* *the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3558–3567, Florence, Italy. Association for Computational Linguistics.

Tianyu Fan, Xinyao Niu, Yuxiang Zheng, Fengji Zhang, Chengen Huang, Bei Chen, Junyang Lin, and Chao Huang. 2025. Understanding deepresearch via reports. *Preprint*, arXiv:2510.07861.

Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. 2023. Enabling large language models to generate text with citations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6465–6488, Singapore. Association for Computational Linguistics.

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Nan Duan, and Weizhu Chen. 2023. Critic: Large language models can self-correct with tool-interactive critiquing. *arXiv preprint arXiv:2305.11738*.

Rujun Han, Yanfei Chen, Zoey CuiZhu, Lesly Miculicich, Guan Sun, Yuanjun Bi, Weiming Wen, Hui Wan, Chunfeng Wen, Solène Maître, and 1 others. 2025. Deep researcher with test-time diffusion. *arXiv preprint arXiv:2507.16075*.

Rujun Han, Yuhao Zhang, Peng Qi, Yumo Xu, Jenyuan Wang, Lan Liu, William Yang Wang, Bonan Min, and Vittorio Castelli. 2024. RAG-QA arena: Evaluating domain robustness for long-form retrieval augmented question answering. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 4354–4374, Miami, Florida, USA. Association for Computational Linguistics.

Helia Hashemi, Jason Eisner, Corby Rosset, Benjamin Van Durme, and Chris Kedzie. 2024. LLM-rubric: A multidimensional, calibrated approach to automated evaluation of natural language texts. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13806–13834, Bangkok, Thailand. Association for Computational Linguistics.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. 2023. Large language models cannot self-correct reasoning yet. *arXiv preprint arXiv:2310.01798*.

Zhengbao Jiang, Frank Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Active retrieval augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7969–7992, Singapore. Association for Computational Linguistics.

LangChain. 2025. Open deep research.

Young-Jun Lee, Seungone Kim, Byung-Kwan Lee, Minkyeong Moon, Yechan Hwang, Jong Myoung Kim, Graham Neubig, Sean Welleck, and Ho-Jin Choi. 2025a. Refinebench: Evaluating refinement capability in language models. In *First Workshop on Multi-Turn Interactions in Large Language Models*.

Yukyung Lee, JoongHoon Kim, Jaehee Kim, Hyowon Cho, Jaewook Kang, Pilsung Kang, and Najoung Kim. 2025b. CheckEval: A reliable LLM-as-a-judge framework for evaluating text generation using checklists. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 15782–15809, Suzhou, China. Association for Computational Linguistics.

Minghao Li, Ying Zeng, Zhihao Cheng, Cong Ma, and Kai Jia. 2025a. Reportbench: Evaluating deep research agents via academic survey tasks. *arXiv preprint arXiv:2508.15804*.

Xiaoxi Li, Jiajie Jin, Guanting Dong, Hongjin Qian, Yutao Zhu, Yongkang Wu, Ji-Rong Wen, and Zhicheng Dou. 2025b. Webthinker: Empowering large reasoning models with deep research capability. *arXiv preprint arXiv:2504.21776*.

Nelson Liu, Tianyi Zhang, and Percy Liang. 2023. Evaluating verifiability in generative search engines. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7001–7025, Singapore. Association for Computational Linguistics.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-refine: Iterative refinement with self-feedback. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Grégoire Mialon, Clémentine Fourrier, Craig Swift, Thomas Wolf, Yann LeCun, and Thomas Scialom. 2023. Gaia: a benchmark for general ai assistants. *Preprint*, arXiv:2311.12983.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candes, and Tatsunori Hashimoto. 2025. s1: Simple test-time scaling. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 20286–20332, Suzhou, China. Association for Computational Linguistics.

Deepak Nathani, David Wang, Liangming Pan, and William Wang. 2023. Maf: Multi-aspect feedback for improving reasoning in large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6591–6616.

OpenAI. 2025. Deep research system card.

Perplexity. 2025. Introducing perplexity deep research.

Long Phan, Alice Gatti, Ziwen Han, Nathaniel Li, Josephina Hu, Hugh Zhang, Chen Bo Calvin Zhang, Mohamed Shaaban, John Ling, Sean Shi, Michael Choi, Anish Agrawal, Arnav Chopra, Adam Khoja, Ryan Kim, Richard Ren, Jason Hausenloy, Oliver Zhang, Mantas Mazeika, and 1093 others. 2025. Humanity's last exam. *Preprint*, arXiv:2501.14249.

Zile Qiao, Guoxin Chen, Xuanzhong Chen, Donglei Yu, Wenbiao Yin, Xinyu Wang, Zhen Zhang, Baixuan Li, Huifeng Yin, Kuan Li, and 1 others. 2025. Webresearcher: Unleashing unbounded reasoning capability in long-horizon agents. *arXiv preprint arXiv:2509.13309*.

Jie Ruan, Inderjeet Nair, Shuyang Cao, Amy Liu, Sheza Munir, Micah Pollens-Dempsey, Tiffany Chiang, Lucy Kates, Nicholas David, Sihan Chen, Ruxin Yang, Yuqian Yang, Jasmine Gump, Tessa Bialek, Vivek Sankaran, Margo Schlanger, and Lu Wang. 2025. Expertlongbench: Benchmarking language models on expert-level long-form generation tasks with structured checklists. *Preprint*, arXiv:2506.01241.

Rulin Shao, Akari Asai, Shannon Zejiang Shen, Hamish Ivison, Varsha Kishore, Jingming Zhuo, Xinran Zhao, Molly Park, Samuel G Finlayson, David Sontag, and 1 others. 2025. Dr tulu: Reinforcement learning with evolving rubrics for deep research. *arXiv preprint arXiv:2511.19399*.

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 M. Hendryx, Brad Kenstler, and Bing Liu. 2025. Researchrubrics: A benchmark of prompts and rubrics for evaluating deep research agents. *Preprint*, arXiv:2511.07685.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik R Narasimhan, and Shunyu Yao. 2023. Reflexion: language agents with verbal reinforcement learning. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Charlie Victor Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2025. Scaling LLM test-time compute optimally can be more effective than scaling parameters for reasoning. In *The Thirteenth International Conference on Learning Representations*.

Yixiao Song, Yekyung Kim, and Mohit Iyyer. 2024. VeriScore: Evaluating the factuality of verifiable claims in long-form text generation. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 9447–9474, Miami, Florida, USA. Association for Computational Linguistics.

Tongyi. 2025. Tongyi deepresearch technical report. *arXiv preprint arXiv:2510.24701*.

Manya Wadhwa, Xinyu Zhao, Junyi Jessy Li, and Greg Durrett. 2024. Learning to refine with fine-grained natural language feedback. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 12281–12308, Miami, Florida, USA. Association for Computational Linguistics.

Jiayu Wang, Yifei Ming, Riya Dulepet, Qinglin Chen, Austin Xu, Zixuan Ke, Frederic Sala, Aws Albarghouthi, Caiming Xiong, and Shafiq Joty. 2025. Liveresearchbench: A live benchmark for user-centric deep research in the wild. *Preprint*, arXiv:2510.14240.

Jason Wei, Zhiqing Sun, Spencer Papay, Scott McKinney, Jeffrey Han, Isa Fulford, Hyung Won Chung, Alex Tachard Passos, William Fedus, and Amelia Glaese. 2025a. Browsecomp: A simple yet challenging benchmark for browsing agents. *arXiv preprint arXiv:2504.12516*.

Tianjun Wei, Wei Wen, Ruizhi Qiao, Xing Sun, and Jianghong Ma. 2025b. Rocketeval: Efficient automated LLM evaluation via grading checklist. In *The Thirteenth International Conference on Learning Representations*.

Tianze Xu, Pengrui Lu, Lyumanshan Ye, Xiangkun Hu, and Pengfei Liu. 2025. Researcherbench: Evaluating deep ai research systems on the frontiers of scientific inquiry. *Preprint*, arXiv:2507.16280.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*.

Yang Yao, Yixu Wang, Yuxuan Zhang, Yi Lu, Tianle Gu, Lingyu Li, Dingyi Zhao, Keming Wu, Haozhe Wang, Ping Nie, Yan Teng, and Yingchun Wang. 2025. A rigorous benchmark with multidimensional evaluation for deep research agents: From answers to reports. *Preprint*, arXiv:2510.02190.

Xi Ye, Ruoxi Sun, Sercan Arik, and Tomas Pfister. 2024. Effective large language model adaptation for improved grounding and citation generation. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6237–6251, Mexico City, Mexico. Association for Computational Linguistics.

Eric Zelikman, Eliana Lorch, Lester Mackey, and Adam Tauman Kalai. 2024. Self-taught optimizer (stop): Recursively self-improving code generation. In *First Conference on Language Modeling*.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023. Instruction-following evaluation for large language models. *Preprint*, arXiv:2311.07911.

A Data Curation Details

A.1 Dataset Details

We show the dataset statistics of the three datasets of MR DRE in Table 4.

Note that for RigorousBench, we used their "Query-Specific Rubrics" as our question-specific checklist, and excluded the more coarse scoring checklist named "General-Report Rubrics" in (Yao et al., 2025). They labeled the score for satisfying each checklist criterion, which we use as the question weight. For all our experiments, we used a sub-sampled set of 100 questions in RigorousBench due to the high cost of Deep Research experiments, which is a sufficient size comparable to the other two datasets.

Dataset	Size	# Items / Question	Domains
ResearchRubrics	101	25.67	General
RigorousBench	214	14.32	General
ResearcherBench	65	13.33	AI&ML

Table 4: **Dataset statistics**. # Items / Question means the average number of instance-specific checklist items of each question.

A.2 Core Set Construction

We construct a core set by uniformly sampling 25 questions from each dataset (ResearcherBench, ResearchRubrics, and RigorousBench) such that for each sampled question, the initial report produced by every evaluated DRA fails to satisfy at least four evaluation criteria.

B Evaluation Protocol Details

Below, we describe details of our evaluation protocol. For all LLM judges ($\mathcal{J}_{\text{cov}}$, $\mathcal{J}_{\text{fact}}$, $\mathcal{J}_{\text{pres}}$) and the claim extractor mentioned in §3, we instantiate with GPT-4.1-mini (`gpt-4.1-mini-2025-04-14`) following DR Tulu, which is strong at instruction-following and long-context understanding, balancing accuracy and cost. We use temperature= 0 for all judgements to minimize randomness and promote reproducibility.

B.1 Comprehensiveness Evaluation

For comprehensiveness evaluation, we evaluate the checklist coverage score using an LLM judge $\mathcal{J}_{\text{cov}}$ for each report-criterion pair. The prompt template is presented in Figure 14.

Note that we append a small reminder text (Figure 5) to the user message for evaluating negative-weight criteria. This is because the LLM Judge sometimes scores 1 when the report avoids what the criterion asks about, which is the opposite of what we expect it to do.

Negative-weight Reminder Text

```
Note: this is a negative criterion. Your
score should be 1.0 only if it describes
something that is present or true to the
report. If the report did not contain the
content described by the criterion, your
score should be 0.0.
```

Figure 5: **Negative-weight Reminder Text.**

B.2 Factuality Evaluation

Here, we describe the detailed pipeline for factuality evaluation. We first split each report into sections using double new lines. Then, for each section, we extract atomic claims alongside their cited URL(s) with the claim extractor LLM (prompt template shown in Figure **??**&**??**). For each claim, we gather its cited URLs and fetch the URL content using the Jina Reader API[2]. To reduce the cost of running LLM judges, we summarize the crawled URL content with a lightweight model, GPT-4.1-Nano, to reduce context length. We then prompt $\mathcal{J}_{\text{fact}}$ to label each claim as `Supported`, `Contradicted`, or `Insufficient`, given the crawled URL content (Prompt in Figure **??**&**??**). As introduced in §3.2, we report citation faithfulness as the fraction of supported claims among claims with at least one URL, and claim groundedness as the fraction of supported claims among all extracted claims.

B.3 Presentation Evaluation

In Table 5, we show the full list of our presentation questions, each with its detailed source or rationale for inclusion. These questions are carefully consolidated and refined from LiveResearchBench (Wang et al., 2025), RigorousBench (Yao et al., 2025), DeepResearch-ReportEval (Fan et al., 2025), with some new questions that we find missing from all prior works. Note that p_6 and p_7 are questions that might not apply to some reports; we exclude them if the judgement score is -1.

[2] `https://jina.ai/reader/`

	Question	Source/Rationale
p_1	Does the report follow a clear, logically ordered structure that is easy to navigate (e.g., problem → approach → results), with sections that match the report's stated purpose and directly address the research question?	Q1 in LiveResearchBench's Table 3; GRR 1 and 2 in RigorousBench; Definition of "Clear and Logical Structure" in DeepResearch-ReportEval
p_2	Do different sections logically follow or build on one another with minimal redundant restatement, and is any repetition clearly purposeful (e.g., brief recap before a new stage)?	Definition of "Redundancy" in DeepResearch-ReportEval; Q2 in LiveResearchBench's Table 3; Refined so that recap/summary is not counted as redundancy
p_3	Where content is naturally parallel (steps, criteria, comparisons, key takeaways), does the report use lists and/or tables to present it in a scannable form rather than dense prose?	Newly written. This is not present in any previous evaluations, but is essential for penalizing dense paragraphs without proper formatting.
p_4	Are headings/subheadings consistent in level and hierarchy (H1/H2/H3), and are comparable sections named with parallel phrasing (e.g., "Method," "Results" rather than inconsistent mixes like "How they did it," "Findings")?	Further specified GRR8, 42 in RigorousBench. Added that the heading names should be parallel and comparable.
p_5	Does the report use concise transition sentences/phrases to signal why the subsequent content follows and to reduce abrupt jumps and make the report easier to follow?	Further specified GRR7 in RigorousBench and Definition of "Clear and Logical Structure" in DeepResearch-ReportEval.
p_6	If there are cross-references, are they consistent and unambiguous (figure/table numbers, section references, in-text citation), with no missing/duplicate numbering and no "see above/below" without an anchor? If no cross-references are present, the score should be -1.	Extended Q3, 4, 6, 10 in LiveResearchBench's Table 3 and GRR24 in RigorousBench to all types of cross-references.
p_7	If tables are included, are they structurally complete and interpretable on their own (no blank cells without notation, consistent units/precision, clear headers/labels/notes)? If no tables are included, the score should be -1.	Extended Q8 in LiveResearchBench's Table 3; We do not allow automatic pass but rather discard it if the report does not have tables
p_8	Is report formatting correct and consistent (e.g., valid Markdown heading syntax, renderable Markdown tables, consistent numbering, consistent emphasis/code styling, consistent citation format if used)?	Further specified and extended Q9 in LiveResearchBench's Table 3
p_9	Is the writing clear and professional at the sentence level (consistent tense/voice, minimal colloquialisms, avoids rhetorical exaggeration), with consistent terminology and abbreviation handling (define once, then reuse consistently)?	Further specified Q9 in LiveResearchBench's Table 3 and GRR48 in RigorousBench. Rewritten so that professionalism is defined more clearly.
p_{10}	Are key terms, symbols, and abbreviations formatted consistently (e.g., italicization, capitalization, acronym, bolding), and is there no drifting where the same concept is labeled multiple ways without intent?	Newly written. Stylistic considerations are missing from previous evaluations, which is important for report presentation.

Table 5: **Presentation Evaluation Questions**

B.4 Handling Negative Weights in Evaluation Metrics

Some Deep Research benchmarks, notably ResearchRubrics (Sharma et al., 2025) in our evaluation suite, may assign negative weights to certain checklist criteria to penalize undesirable content such as misinformation or irrelevant topics. For a criterion c_i with negative weight $w_i < 0$, a score of $s_i = 1$ (full coverage) indicates the report contains the undesirable content and should be penalized, while $s_i = 0$ indicates the report correctly avoids it. Below we describe how each metric accommodates negative weights. The actual results in our experiment sections use the formulations below instead of the simplified version in §3 without considering negative weights.

B.4.1 Coverage Score

For the coverage score, the numerator $\sum_{i=1}^{n} w_i \cdot s_i$ naturally handles negative weights: when $w_i < 0$ and $s_i > 0$, the product $w_i \cdot s_i$ is negative, reducing the overall score. However, the denominator must be adjusted to normalize only by the maximum achievable score, which comes from positive-weight criteria alone (since the best outcome for negative-weight criteria is $s_i = 0$, contributing nothing to the numerator). The full formula is:

$$\textsc{Cov}(r) = \frac{\sum_{i=1}^{n} w_i \cdot s_i}{\sum_{i:w_i>0} w_i}$$

This formulation ensures that the coverage score ranges from negative values (when the report contains penalized content) to 1 (when the report fully covers all positive-weight criteria and avoids all negative-weight criteria).

B.4.2 Incorporation Rate

For content feedback, the incorporation rate measures whether feedback targets reach their ideal coverage score after revision. The ideal score depends on the sign of the weight:

$$\bar{s}_i = \begin{cases} 1, & \text{if } w_i > 0 \\ 0, & \text{if } w_i < 0 \end{cases}$$

For positive-weight criteria, the ideal is full coverage ($s_i = 1$). For negative-weight criteria, the ideal is zero coverage ($s_i = 0$), meaning the report should remove or avoid the undesirable content. The incorporation rate at turn t becomes:

$$\textsc{Inc} = \frac{1}{|\mathcal{T}^{(t)}|} \sum_{c_i \in \mathcal{T}^{(t)}} \mathbb{1}\left[s_i^{(t)} = \bar{s}_i\right]$$

Note that feedback targets $\mathcal{T}^{(t)}$ are sampled from criteria that have not yet reached their ideal score, which for negative-weight criteria means $s_i^{(t-1)} > 0$.

B.4.3 Break Rate

The break rate measures the degradation of previously achieved coverage. The definitions of "previously achieved" and "coverage degradation" are adapted based on the weight sign:

Previously Achieved Coverage For positive-weight criteria, previously achieved coverage means $s_i^{(t-1)} > 0$ (at least partial coverage of desirable content). For negative-weight criteria, previously achieved coverage means $s_i^{(t-1)} < 1$ (not fully covering undesirable content, i.e., partially or fully avoiding the misconception). Let $\mathcal{C}_+^{(t-1)}$ denote the set of criteria with previously achieved coverage:

$$\mathcal{C}_+^{(t-1)} = \{c_i : (w_i > 0 \wedge s_i^{(t-1)} > 0) \vee (w_i < 0 \wedge s_i^{(t-1)} < 1)\}$$

Coverage Degradation For positive-weight criteria, degradation occurs when coverage decreases ($s_i^{(t)} < s_i^{(t-1)}$). For negative-weight criteria, degradation occurs when coverage increases ($s_i^{(t)} > s_i^{(t-1)}$), meaning the revision introduced more undesirable content. Both cases can be unified using the weight sign: degradation occurs when $w_i \cdot s_i^{(t)} < w_i \cdot s_i^{(t-1)}$, i.e., when the weighted contribution to the coverage score decreases. The full break rate formula is:

$$\textsc{Brk} = \frac{|\{c_i \in \mathcal{C}_+^{(t-1)} : w_i \cdot s_i^{(t)} < w_i \cdot s_i^{(t-1)}\}|}{|\mathcal{C}_+^{(t-1)}|}$$

B.5 Human Evaluation of Coverage LLM Judge

To further validate the reliability of the LLM judge and Break Rate metric in Mr Dre's evaluation protocol, we conduct a human evaluation: We randomly sampled 30 criteria that were judged as satisfied in Turn 1 but broken in Turn 2, drawn from all five DRAs' generations across three benchmarks and under both the Content_1 and Format feedback settings. In total, we gather 60 (question, report, checklist criterion, LLM judgment) tuples. Two of our authors annotated whether they agreed with the LLM judge's assessments, scoring each as agree (1.0), partial agree (0.5), or disagree (0.0). We report the LLM-human agreement rate for Turn 1 and Turn 2 reports separately and collectively in Table 6.

	LLM-Human Agreement Rate
Turn 1	95.0%
Turn 2	96.7%
All	95.8%

Table 6: **LLM Judge-human agreement rate across turns.**

We observe that agreement rates are high ($\geq$ 95%) on both turns, confirming the reliability of our LLM judge and the Break Rate metric. Also, the similar agreement rates between Turn 1 and Turn 2 further validate that the judge does not perform differently on revised (potentially paraphrased) content versus the original.

C Feedback Simulation Pipeline Details

C.1 Content Feedback

To simulate content feedback, we prompt GPT-4.1-mini with the question, k sampled feedback targets with each score, weight, and scoring justification. We show the prompt for $k = 1$ in Figure 11 and $k > 1$ in Figure 13.

C.2 Seed Format Feedback

Two of our authors wrote the following 21 diverse and realistic seed format feedback pieces. We present them in Table 7.

C.3 Human Validation Results

Our goal is to simulate the most realistic follow-up that a human user would ask the DRA to revise the report against. Therefore, we defined the following four dimensions to assess the feedback's quality:

Naturalness: The language and wording should be natural and human-like, as if it were a natural follow-up response from the user themselves, or a thoughtful peer/supervisor.

Draft-specificness: The feedback should be tailored to the question and the current draft of the report, targeting aspects that the current draft misses and have clear room for improvement.

Actionability: The feedback should be concrete and actionable, phrased as implementable suggestions and avoiding vague comments such as "improve clarity" without explaining how.

Content-preserving (only applicable to format feedback): The feedback must not require any edits to existing content in the current draft. It should only incur changes in the form, structure, organization, tone, or style of the writing.

From all $Content_1$ and Format feedback generated for five DRAs across three datasets, we randomly sampled 50 content feedback instances and 50 format feedback instances. Two authors, each holding at least a Bachelor's degree in a science-related field, independently annotated each feedback instance alongside its corresponding report on the four dimensions above using binary scores (`satisfied` or `not satisfied`). We report agreement rate as the percentage of instances where both annotators assign identical scores across all four dimensions. For instances with disagreement, we take the lower score to provide a conservative estimate of feedback quality. We present the results in Table 8.

We found that our feedback simulation pipeline generally achieves a near-perfect score across all dimensions with a high inter-annotator agreement rate. This validates our feedback simulation pipeline as a realistic component for multi-turn report revision.

D Proposed Fixes Details

D.1 Prompt Engineering (PE) on Feedback Details

The prompt engineering (PE) fix pipeline refines raw user feedback into an executable revision instruction in two steps. First, we feed the original query, the full research report, and the user's feedback into a prompt refiner (GPT-4.1) with a fixed system prompt (Figure 26) that forces the output into a structured, localized edit plan. Second, we append a fixed, hard-coded constraint suffix to this structured plan, which makes the downstream editor follow only the specified actions, avoid global rewrites, and output only the revised report. The concatenation of the structured edit plan and the constraint suffix (Figure 27) forms the final refined prompt used for report revision.

D.2 Reviser Subagent Details

For the Reviser, we implemented a simple ReAct agent using Qwen3-30B-A3B-Instruct-2507 as the backbone model and its default function calling template, augmented with Serper API[3] to call

[3]https://serper.dev/

ID	Feedback
1	Please rewrite this so the language is clearer and more straightforward, suitable for a reader with no prior knowledge.
2	Whenever you introduce a technical concept, add a simple and real-world analogy to illustrate it.
3	Standardize heading levels and naming so similar sections use parallel phrasing (e.g., 'Approach', 'Results', 'Limitations').
4	Make sure that each section ends with a short summary sentence that emphasizes the main takeaway.
5	Add a concise TL;DR at the beginning of the report that states the main question and key takeaways from the report.
6	It would help if the report indicated which parts are essential reading and which parts are optional background.
7	Highlight key sentences or phrases (e.g., with bold) so I can quickly find the most important takeaways.
8	Please add short 'section previews' at the start of each main section, summarizing in 1–2 lines what will be covered.
9	Please keep the core sections concise and move extended explanations, detailed justifications, and long background passages into clearly labeled 'Appendix' sections at the end.
10	Consider adding transition sentences between sections to show how each part connects to the next.
11	Add subheadings every 2-3 paragraphs to help readers navigate and find information quickly.
12	Include a glossary of key terms at the end for readers who want quick reference.
13	Consider using bullet points or numbered lists when presenting multiple related items rather than embedding them in prose.
14	Add visual breaks like pull quotes to highlight critical insights so that it's easier to find takeaways.
15	Apply bold formatting to critical findings, main conclusions, and essential terms on first mention, while using italics for secondary emphasis, technical terms in context, or when citing specific examples.
16	Add a "How to Read This Report" section that explains the document's structure and what different readers should focus on.
17	Vary sentence length and structure to maintain reader interest and create rhythm.
18	Use "we" as much as possible than "you" or third-person pronouns to create connection with readers rather than maintaining complete detachment.
19	Add a brief "Why This Matters" box at the start of technical sections to motivate readers.
20	Close with actionable next steps or recommendations for related information so readers know what to do or read next.
21	Create a separate "Frequently Asked Questions" section to address common points of confusion.

Table 7: 21 Seed Format Feedback Pieces in MR DRE.

	Content	Format
Naturalness	100%	100%
Draft-specificness	92%	90%
Actionability	98%	98%
Content-preserving	–	100%
Agreement Rate	98%	96%

Table 8: **Human verification results of simulated feedback quality.**

Google Search for additional information when the user feedback requires some extra information gathering. We set the temperature to 0.7, top-p to 0.95, and the maximum number of generated tokens to 16384. For each revision, we allow the agent to call the search API 10 times at maximum, with each call returning the top 5 web pages. If the maximal number of tool calls is reached, we softly force a final answer by using "`You have reached the maximal number of web search calls. Please now produce the revised report based on the information you have and the user feedback.`" as the tool output. The system and user prompt templates are in Figure 28 and 29.

E Additional Results

E.1 Citation Analysis

We present citation-related statistics in Table 13, reporting the average number of extracted claims ($|\mathcal{E}|$), claims with at least one citation ($|\mathcal{E}_{\text{cited}}|$), supported claims ($|\mathcal{S}|$), and citation counts ($|\mathcal{U}|$) for each dataset. These fine-grained statistics reveal that the causes of citation degradation vary across agents.

For **OpenAI DR**, degradation stems primarily from reductions in both supported claims and overall citation counts, with supported claims declining more severely (on average -7.0 supported claims and -11.2 citation counts).

For **Sonar DR**, as noted in Section 5.1, 68% of reports generated after self-reflection contain zero citations, causing both factuality metrics to plummet. A similar pattern emerges in the Format setting, where 21% of reports on average lack any cited URLs. While this phenomenon disappears for ResearchRubrics and ResearcherBench under Content$_1$, we still observe a substantial reduction in the ratio of supported claims.

For **LC ODR**, the Reflect setting produces notably more claims than the initial draft, yet the number of supported claims does not increase pro-

Setting	ResearchRubric (10 samples) Cov.	Fa.	Gr.	Pre.	Inc.	Brk.
Initial	76.0	59.0	24.1	98.9	–	–
Reflect	+0.5	-21.7	-11.5	0.0	–	4.6
Content_1	-11.3	-4.6	0.0	-3.3	80.0	19.1
Format	-2.4	-14.3	-9.7	+1.1	100.0	8.4

Table 9: **`o3-deep-research` results on the ResearchRubric subset (10 samples).** Second turn's score change from Initial (turn 1) is shown for the four main metrics, along with incorporation (Inc.) and break (Brk.) rates.

portionally, leading to lower citation faithfulness and claim groundedness. In the Content_1 and Format settings, claim counts remain relatively stable or increase slightly, but the number of cited or supported claims drops, yielding similar degradation in citation quality.

For **DR Tulu**, the relatively stable but mixed number of supported claims across settings explains why it exhibits the most consistent citation quality among all evaluated DRAs.

E.2 Full Multi-turn Results under Content_1 and Reflect

We present the complete results of Content_1 and Reflect up to 4 turns of revision in Figure 7.

E.3 Full Multi-item Content Feedback Results

We present the complete results of Content_k with a varying number of feedback targets in Figure 8.

E.4 Model Scaling Results with `o3-deep-research`

In our main experiments, we use `o4-mini-deep-research` for OpenAI Deep Research results, while a supposedly more capable model `o3-deep-research` is available. It is also interesting to understand how model scaling affects multi-turn revision abilities of DRAs. However, we are unable to conduct this experiment due to its prohibitive cost: a single report costs an estimated $5–8. Evaluating multi-turn revision requires three feedback settings and at least two turns of generation, which would cost up to $2,400 even on our sub-sampled Core Set.

Nevertheless, to directly address whether a stronger backbone model can resolve the unreliability in multi-turn report revision, we provide some preliminary results running `o3-deep-research` on 10 randomly sampled questions from ResearchRubrics in Table 9.

We found that even `o3-deep-research` exhibits coverage and citation degradation across three feedback settings, with a notable 19.1% break rate under content feedback. We acknowledge that 10 samples can be small and the scores are noisy, making them incomparable with the main table. Yet, these results suggest that *scaling the backbone model alone is unlikely to fully resolve the multi-turn revision unreliability*.

E.5 Statistical Significance Tests for Main Observations

Given that the size of Deep Research benchmarks is generally small, we conduct *one-sided paired t-tests* on per-question paired differences in our experiments. The one-sided paired t-test computes the per-question difference between two conditions (e.g., coverage score before vs. after revision) and tests whether the mean difference is significantly below or above zero. We test three key observations in our experiment results and report p-values for each DRA.[4]

Obs 1: Coverage score decreases after revision for all feedback settings (Turn 1 → Turn 2). This is observed on three datasets in Table 2. We test H_0: coverage score change ≥ 0 vs. H_1: coverage score change < 0 on results pooled from all three datasets. Results are shown in Table 10.

Agent	Content_1	Format	Reflect
OpenAI DR	<0.001**	<0.001**	<0.001**
Soner DR	<0.001**	<0.001**	<0.001**
LC ODR	<0.001**	<0.001**	1.0
Tongyi DR	<0.001**	<0.001**	0.315
DR Tulu	0.565	0.002**	0.059

Table 10: p-values for Obs 1 (coverage decrease after revision).

Obs 2: Break rate is significant for all DRAs over multiple turns of revision. This is observed in Figure 3 (right). We test H_0: break rate ≤ 0 vs. H_1: break rate > 0 under the Content_1 setting. Results are shown in Table 11.

Obs 3: Actual coverage lags behind oracle over multiple turns of revision. This is observed in Figure 3 (left). We test H_0: actual − oracle ≥ 0 vs. H_1: actual − oracle < 0 under the Content_1 setting. Results are shown in Table 12.

[4] Throughout this section, * denotes $p < 0.05$ and ** denotes $p < 0.01$.

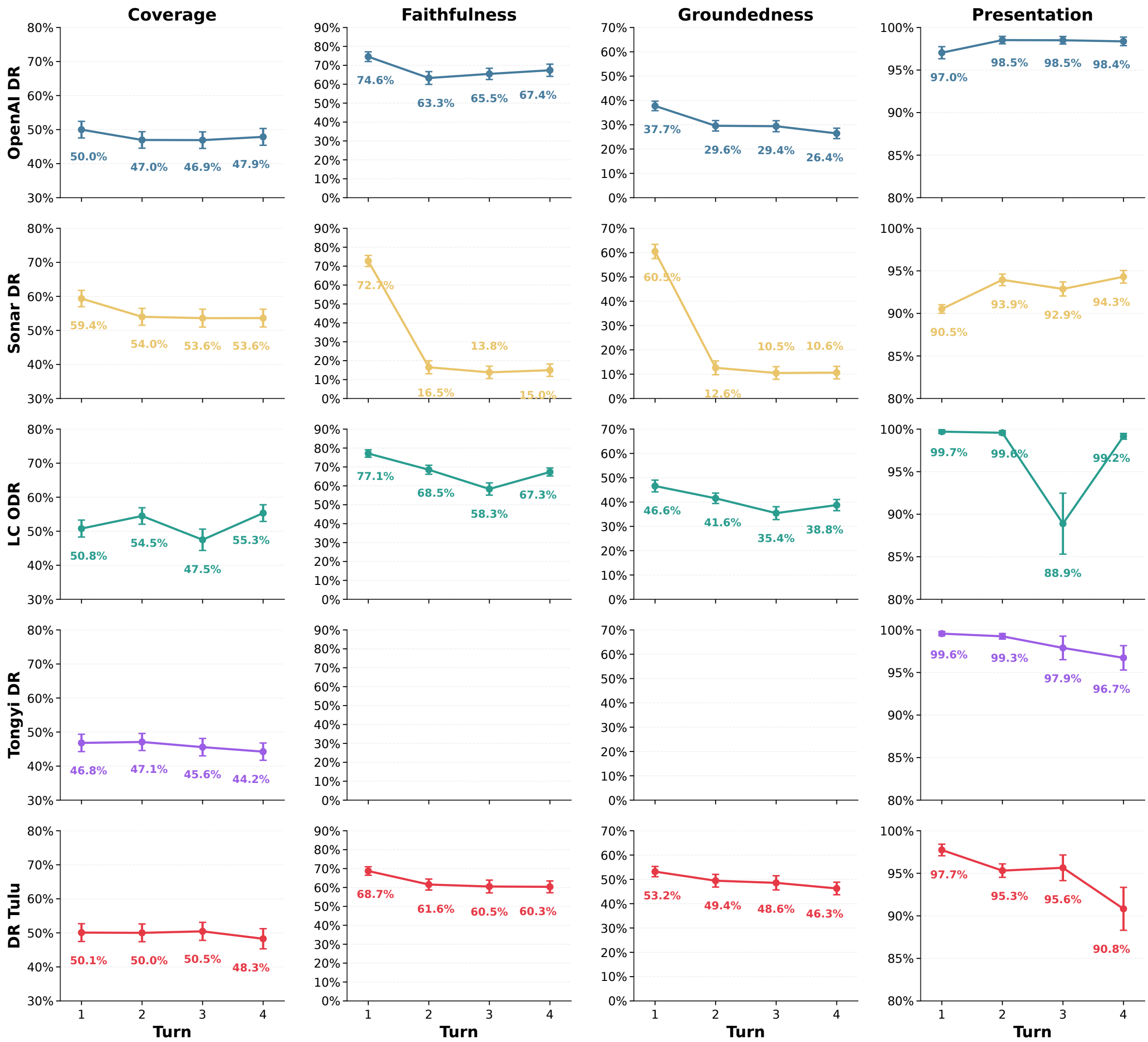

Figure 6: **Full multi-turn results for the Reflect setting.** Tongyi DR's citation faithfulness and claim groundedness are omitted since it is not trained to generate citations. Error bars indicate standard errors.

Agent	Turn 2	Turn 3	Turn 4
OpenAI DR	<0.001**	<0.001**	<0.001**
Soner DR	<0.001**	<0.001**	<0.001**
LC ODR	<0.001**	<0.001**	<0.001**
Tongyi DR	<0.001**	<0.001**	<0.001**
DR Tulu	<0.001**	<0.001**	<0.001**

Table 11: p-values for Obs 2 (break rate over multiple turns).

Agent	Turn 2	Turn 3	Turn 4
OpenAI DR	<0.001**	<0.001**	<0.001**
Soner DR	<0.001**	<0.001**	<0.001**
LC ODR	<0.001**	<0.001**	<0.001**
Tongyi DR	0.003**	<0.001**	<0.001**
DR Tulu	0.022*	<0.001**	<0.001**

Table 12: p-values for Obs 3 (actual coverage vs. oracle).

All three observations are statistically significant at $p < 0.01$ for almost all agents. These results confirm that our findings reflect systematic limitations of current DRAs in multi-turn revision.

F Prompt Templates

We present prompt templates in Figure 11-29.

G Case Studies

G.1 More Feedback Examples

We present representative feedback examples used in our multi-turn revision setup (Table 14). The table includes both format feedback and content feedback with one or three targets.

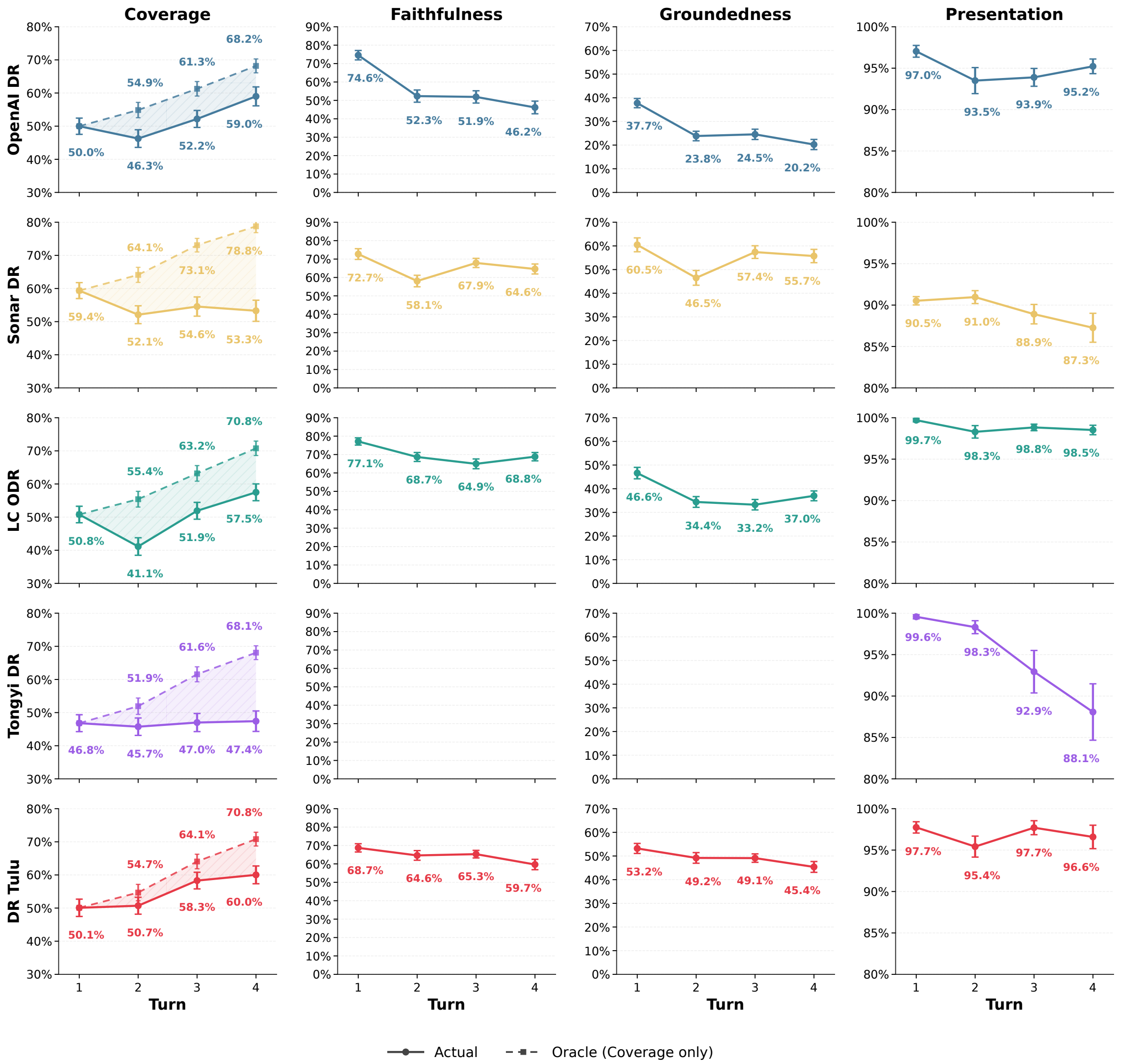

Figure 7: **Full multi-turn results for the Content$_1$ setting.** Tongyi DR's citation faithfulness and claim groundedness are omitted since it is not trained to generate citations. Error bars indicate standard errors.

G.2 Error Cases

We provide two representative failure cases in multi-turn report revision. Figure 9 shows a *missing content* case, where the revised report fails to preserve content outside the feedback's scope and drops a required paragraph. Figure 10 illustrates *citation degradation*, where the revised report reduces citations and removes in-context citation markers from the original.

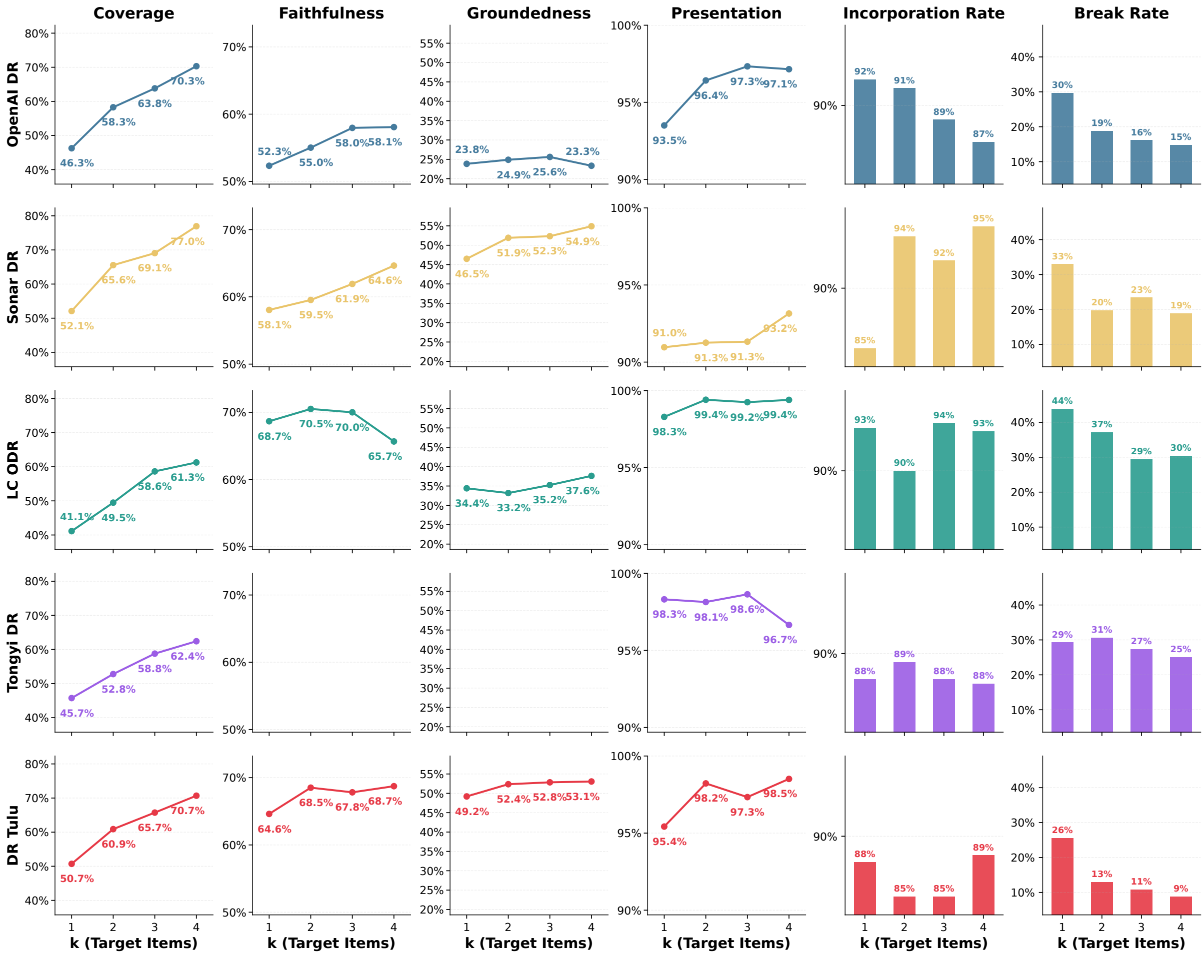

Figure 8: **Full results for content feedback with multiple feedback targets (k).**

Agent	Setting	ResearchRubrics $\lvert\mathcal{E}\rvert$	ResearchRubrics $\lvert\mathcal{E}_{\text{cited}}\rvert$	ResearchRubrics $\lvert\mathcal{S}\rvert$	ResearchRubrics $\lvert\mathcal{U}\rvert$	RigorousBench $\lvert\mathcal{E}\rvert$	RigorousBench $\lvert\mathcal{E}_{\text{cited}}\rvert$	RigorousBench $\lvert\mathcal{S}\rvert$	RigorousBench $\lvert\mathcal{U}\rvert$	ResearcherBench $\lvert\mathcal{E}\rvert$	ResearcherBench $\lvert\mathcal{E}_{\text{cited}}\rvert$	ResearcherBench $\lvert\mathcal{S}\rvert$	ResearcherBench $\lvert\mathcal{U}\rvert$	Avg $\lvert\mathcal{E}\rvert$	Avg $\lvert\mathcal{E}_{\text{cited}}\rvert$	Avg $\lvert\mathcal{S}\rvert$	Avg $\lvert\mathcal{U}\rvert$
OpenAI DR	Init	73.3	29.5	20.4	15.1	73.8	36.1	23.7	19.1	44.8	22.2	17.7	11.4	64.0	29.3	20.6	15.2
	Reflect	-2.8	-8.3	-6.8	-2.6	-2.4	-4.6	-3.8	-1.2	-0.5	-3.2	-2.8	-3.3	-1.9	-5.3	-4.5	-2.4
	Content_1	-7.2	-6.1	-8.3	-7.2	-11.6	-7.1	-11.3	-8.0	-6.8	-5.8	-7.9	-4.7	-8.5	-6.3	-9.2	-6.6
	Format	+2.4	-8.6	-9.6	-7.2	-4.9	-7.5	-8.0	-9.2	-0.9	-3.2	-4.4	-3.7	-1.1	-6.4	-7.3	-6.7
Sonar DR	Init	148.5	116.5	85.1	30.1	162.8	136.6	105.5	33.4	148.0	124.2	98.4	32.3	153.1	125.8	96.4	31.9
	Reflect	-22.0	-95.2	-74.9	-21.5	-25.3	-107	-92.4	-22.6	-43.8	-106	-93.6	-25.6	-30.4	-101	-87.0	-23.2
	Content_1	+0.2	-1.0	-28.9	-3.2	-25.7	-31.6	-49.0	-7.5	+3.2	+9.6	-10.2	-0.3	-7.4	-7.7	-29.4	-3.6
	Format	-30.6	-35.0	-51.0	-9.8	-40.5	-46.7	-62.6	-8.6	-24.0	-26.6	-48.6	-4.7	-31.7	-36.1	-54.0	-7.7
LC ODR	Init	64.3	33.7	24.6	16.3	61.4	36.5	26.8	19.1	60.4	37.0	30.4	19.6	62.0	35.7	27.3	18.3
	Reflect	+10.5	+4.4	+1.2	+2.3	+11.6	+8.6	+5.4	+1.9	+9.6	+1.4	-1.7	-1.3	+10.6	+4.8	+1.7	+0.9
	Content_1	+1.8	-4.4	-4.3	-3.5	+0.2	-2.9	-3.8	-3.5	+2.7	-9.2	-10.2	-5.9	+1.6	-5.5	-6.1	-4.3
	Format	+1.6	-1.2	+0.3	-1.2	+3.2	-3.6	-3.2	-3.7	-0.1	-8.9	-9.0	-3.1	+1.6	-4.6	-3.9	-2.6
DR Tulu	Init	93.7	65.8	43.2	18.6	89.4	68.5	42.5	19.1	70.2	54.3	43.2	18.5	84.5	62.9	43.0	18.7
	Reflect	-1.7	+0.9	-1.3	+0.0	+0.4	+1.6	+0.3	+0.4	-2.2	-2.5	-4.0	-1.7	-1.2	+0.0	-1.6	-0.4
	Content_1	+1.0	+2.9	-0.9	+2.7	+1.4	+3.9	-0.5	+5.3	+5.9	+1.9	-0.5	+3.2	+2.8	+2.9	-0.6	+3.8
	Format	-0.6	+3.9	+2.9	+0.4	+4.0	+6.3	+5.1	+2.1	-0.7	-0.4	-2.4	-1.0	+0.9	+3.3	+1.8	+0.5

Table 13: **Full Citation-related Results.** For each dataset, we report the average number of extracted claims ($\lvert\mathcal{E}\rvert$), claims with at least one citation ($\lvert\mathcal{E}_{\text{cited}}\rvert$), supported claims ($\lvert\mathcal{S}\rvert$), and citation counts ($\lvert\mathcal{U}\rvert$). Avg is the four counts averaged across all samples in three datasets.

Feedback type	Feedback example
Format	Including a glossary of key terms at the end of the report would greatly benefit beginners by providing a quick reference to important concepts like forward propagation, backpropagation, and optimization methods, helping to reinforce understanding as they read through the material.
Format	Consider applying bold formatting to key technical terms, main product features, and critical advantages when they first appear, while using italics for secondary details like specific APIs or model names; this will help readers quickly identify the most important information and improve overall readability.
Content_1	Thanks for covering the MYC pathway—it's a great start! To make the overview stronger, could you also include how NELF-E affects other important genes or pathways like BRCA1, RAD51, or its role in promoter-proximal pausing in HCC? That would really round out the explanation.
Content_1	Hey, could you add a part explaining how S1 shows that smaller, high-quality datasets can match the performance of much larger ones? Right now, it mostly talks about large-scale data but misses that important insight about data quality over quantity.
Content_3	Hey, the report would be way stronger if it included a clear marketing strategy covering at least four channels like social media influencers, local events, digital ads, and food delivery promos, with a quick note on how each helps build the brand. Also, it's important to add staffing details for each concept—like specific roles needed for the food truck, fine dining, and fast casual spots—so we get a better sense of the team structure. Lastly, while naming a couple of design firms was helpful, including their contact info and some rough cost estimates would make it easier to move forward and compare options.

Table 14: **More feedback examples.** We show representative format feedback and content feedback with one or three targets.

BEFORE FEEDBACK:

1. Architectural and Training Innovations

1.1 Mixture-of-Experts Architecture with Dynamic, Balanced Routing

DeepSeek V3, an open-source LLM with 671 billion parameters but activating only 37 billion per inference, introduced significant architectural advances that have influenced LLM design. Mixture-of-Experts (MoE) Architecture: V3 uses a hybrid MoE model with shared and routed experts. Multi-Head Latent Attention (MLA): MLA compresses key-value vectors into low-rank latent spaces. Long-Context Support: DeepSeek V3 can handle sequences up to 128,000 tokens (in later versions). Hybrid Reasoning Architecture (V3.1): introduced toggling between fast inference and deep chain-of-thought processing via token-controlled templates. Sparse Attention Mechanism (V3.2): DeepSeek Sparse Attention (DSA) combined with MoE lowers attention computation complexity from $O(L^2)$ to near-linear $O(Lk)$. These innovations collectively advance the architectural state-of-the-art by improving scalability, performance, and inference efficiency over leading proprietary models such as GPT-4o and Claude 3.5, particularly excelling in coding and mathematical reasoning benchmarks [1][6][7][9][12][18][34].

...

[1] Why DeepSeek v3 matters in the world of LLMs – Kiseki Labs: `https://www.kisekilabs.com/blog-posts/why-deepseek-v3-matters-in-the-world-of-llms`

[2] DeepSeek-R1 Release: `https://api-docs.deepseek.com/news/news250120`

...

[34] A Technical Tour of the DeepSeek Models from V3 to V3.2 – Sebastian Raschka Magazine: `https://magazine.sebastianraschka.com/p/technical-deepseek`

FEEDBACK:

The report could be improved by explicitly highlighting R1's standout creative task performance and diving deeper into the specialized domain optimization strategies it uses, as these details really show why it's a strong open-source alternative. Adding these points would give a clearer picture of its unique strengths beyond just benchmark scores.

AFTER FEEDBACK:

1. Architectural and Training Innovations

1.1 Mixture-of-Experts Architecture with Dynamic, Balanced Routing

DeepSeek R1 uses a massive Mixture-of-Experts (MoE) transformer structure hosting approximately 671 billion parameters. Key architectural features supporting creativity and domain specialty include: Hierarchical Entropy-Gated Routing: R1 introduces a bias-based gating mechanism using sigmoid functions. This multi-level gating prunes and directs tokens precisely to domain-specific expert subnetworks. Device-Constrained Expert Allocation: experts are allocated and routed considering hardware topology, minimizing inter-device communication overhead, which allows scalable training and inference even with massive model size.

[MISSING] in-context citations

...

[1] DeepSeek R1: Architectural Innovation in Reasoning Models: `https://mbrenndoerfer.com/writing/deepseek-r1-architectural-innovation-reasoning-models`

[2] DeepSeek-R1: Technical Overview of its Architecture and Innovations: `https://www.geeksforgeeks.org/artificial-intelligence/`

...

[19] DeepSeek-R1: Features, o1 Comparison, Distilled Models: `https://www.datacamp.com/blog/deepseek-r1`

Figure 9: **Citation Degradation Example.** After revision, the model reduces the number of citations and omits in-context citations from the original response.

BEFORE FEEDBACK:

A Brief Company Background: From Startup to Tech Titan:

Founded in 2003 by Peter Thiel, Nathan Gettings, Joe Lonsdale, Stephen Cohen, and Alex Karp, Palantir Technologies began life as a startup driven by a singular ambition: building sophisticated software capable of integrating, analyzing, and visualizing huge volumes of disparate data. Its name comes from the "seeing stones" in J.R.R. Tolkien's works—a nod to the company's mission of offering unparalleled insight through data fusion.

Initially, Palantir developed Gotham, a powerful platform customized for intelligence agencies, law enforcement, and national security organizations. Gotham enabled analysts to correlate mass surveillance, crime, and threat intelligence, dramatically increasing their ability to spot patterns invisible to traditional methods. Simultaneously, Palantir created Foundry for commercial clients, focusing on logistics, health care, manufacturing, and finance—providing companies the same analytical muscle as top-tier intelligence services.

From Secrecy to Public Markets: For years, Palantir operated largely behind closed doors, serving U.S. government entities including the CIA, FBI, and Department of Homeland Security. Only in September 2020 did the company go public via a direct listing, triggering a surge of interest from institutional investors and analysts seeking exposure to disruptive data technologies. While early revenue primarily stemmed from government contracts, the IPO marked Palantir's commitment to broaden its reach, signaling that its most valuable innovations would soon impact sectors far beyond security and intelligence.

FEEDBACK:

To enhance reader engagement throughout this detailed article, try varying sentence length and structure more deliberately—mix shorter, punchier sentences with longer, complex ones to create a natural rhythm that keeps the reader's attention from start to finish.

AFTER FEEDBACK:

Origins: From Startup Dreams to National Security Pillar

Founded in 2003 by Peter Thiel, Nathan Gettings, Joe Lonsdale, Stephen Cohen, and Alex Karp, Palantir began as an audacious bet on big data. Its name, inspired by Tolkien's "seeing stones", promised something radical: software that could stitch together massive, messy datasets into clear human insights.

Initially, Palantir courted governments hungry for answers post-9/11. Its Gotham platform became vital to CIA, FBI, and Department of Homeland Security analysts. It linked surveillance feeds, financial records, and travel logs—a digital Sherlock Holmes that spotted threats humans missed. For years, Palantir operated in shadow. Its first clients were classified. Its milestones unannounced. The IPO in September 2020 was its first public debut, yet the secret behind its power had only just begun to surface.

[MISSING] paragraph

Figure 10: **Example of missing content.** The revised response preserves the overall narrative but omits the final paragraph from the original.

Content$_1$ feedback Simulation System Prompt

```
You are a user providing feedback to a research report writing agent.

You will be provided with:
    1. The original query that you asked
    2. A specific evaluation rule that was used to assess the report where the agent achieves a
    suboptimal score
    3. The coverage status (whether the rule was covered in the report)
    4. The weight of the rule (positive means the rule should be covered, negative means the
    rule should NOT be covered)
    5. The evaluator's explanation for the score

Your task is to provide natural, constructive, concrete feedback that a normal user would give
to improve the report based on this specific evaluation rule.

Guidelines:
    - Be converstational and natural as possible. Imagine you are talking to a collaborator who
    is helping you to write the report.
    - Occasionally, you can use some colloquial language to make the feedback more realistic.
    - Your feedback should be in 1-2 sentences that is concise and to the point.
    - Don't repeat the evaluation explanation verbatim but use it as a reference to help you
    provide the feedback.
```

Figure 11: **Content$_1$ feedback Simulation System Prompt.**

Content$_k$ feedback Simulation System Prompt

```
You are a user providing feedback to a research report writing agent.

You will be provided with:
    1. The original query that you asked
    2. Several specific evaluation rules that were used to assess the report where the agent
    achieves suboptimal scores
    3. For each rule, the coverage status (whether the rule was covered in the report)
    4. For each rule, the weight of the rule (positive means the rule should be covered,
    negative means the rule should NOT be covered)
    5. The evaluator's explanation for each score

Your task is to provide natural, constructive, concrete feedback that a normal user would give
to improve the report based on these specific evaluation rules.

Guidelines:
    - Be converstational and natural as possible. Imagine you are talking to a collaborator who
    is helping you to write the report.
    - Occasionally, you can use some colloquial language to make the feedback more realistic.
    - Your feedback for each evaluation point should be in 1-2 sentences that is concise and to
    the point.
    - Don't repeat the evaluation explanations verbatim but use them as a reference to help you
    provide the feedback.
```

Figure 12: **Content$_k$ feedback Simulation System Prompt.**

Format feedback Simulation System Prompt

```
You are a user providing feedback on the report's writing, structure, and presentation only,
not on its facts, reasoning, or conclusions.

You will be provided with:
1. The original query given to the agent
2. The agent-generated research report
3. Three seed feedback examples from a predefined list

Your task is to:
1. First, select which of the three feedback examples would be most suitable and relevant for
improving this report. It should be targeting an aspect that the report misses or did not do
well. Do not give feedback to what is already done in the report. For example, if the report
already uses subheadings or bulleted lists to organize information, you should not give
feedback on that but select another aspect to ask for improvement.
2. Then, start from the selected feedback and either (a) rewrite it to be more specific and
tailored to the actual content of the report while preserving its core suggestion, or (b) if a
slightly different but closely related suggestion would better improve this particular report,
adapt it into that alternative while staying within the same improvement category.

Your final feedback must adhere to the following specific desiderata:
- **Content-preserving**: Your feedback must not require any edits to existing content in the
current draft. It should only incur changes in the form, structure, organization, tone, or
style of the writing. Do NOT ask for new evidence, new arguments, or different conclusions.
- **Naturalness**: The language and wording should be natural and human-like, as if it was a
natural follow-up response from the user themselves, or a thoughtful peer/supervisor. Give
exactly one coherent suggestion (1-2 sentences) that feels like a natural follow-up from the
user.
- **Draft-specific**: The feedback should be tailored to the original query and the current
draft of the report, targeting aspects that the current draft misses and have clear room for
improvement.
- **Actionability**: The feedback should be concrete and actionable, phrasing as implementable
suggestions and avoiding vague comments such as "improve clarity" without explaining how. Make
it specific to this draft and clearly implementable.

Please only respond with the final rewritten feedback, without any additional explanation or
commentary.
```

Figure 13: **Format feedback Simulation System Prompt.**

Checklist Evaluation System Prompt

```
You will be given a question the user asked (in <question></question> tags) and the
corresponding report (in <report></report> tags) given as a response to the question by an
assistant. You will then be given a specific criterion to evaluate the report against (in
<criterion></criterion> tags. It could be a yes/no question or a statement about the report
that you should judge whether it's true or not).

Your task is to score the report based on whether it satisfies the criterion or not on a
three-point scale: 1.0 if the report satisfies the criterion, 0.5 if the report partially
satisfies the criterion, 0.0 if the report does not satisfy the criterion. Judge only the
specified aspect(s), not any other qualities of the report. Please also provide a short (2-3
sentences maximum) justification for your score. Note: A criterion might be positive or
negative. Satisfying the criterion means that the report contains the content that is
described by the criterion, which should not be confused with satisfying the user's request.

Output only a JSON string with the following format: {\"score\": float, \"justification\":
string}. Do not include any other text or comments in your response.
```

Figure 14: **Checklist Evaluation System Prompt**

Checklist Evaluation User Prompt

```
Evaluate the report based on the given criterion.\n
<question>\n{question}\n</question>\n
<report>\n{report}\n</report>\n
<criterion>\n{criterion}\n</criterion>\n
```

Figure 15: **Checklist Evaluation User Prompt.**

Claim Extraction System Prompt

```
You will be provided with a research report (in <report></report> tags). The body of the report
will contain many factual claims and citations to references. A section of the report will be
highlighted in <highlighted_section></highlighted_section> tags.

Your task is to extract all factual claims from and only from this highlighted section, along
with the corresponding citation URLs if they exist.


Extraction Guidelines:
- You should ONLY extract claims from the highlighted section. Other parts of the report should
only be used as context.
- Each of these claims should be verifiable against external sources (e.g., via Wikipedia). Any
story, personal experiences, hypotheticals (e.g.,\"would be\" or subjunctive), subjective
statements (e.g., opinions), suggestions, advice, instructions, and other such content should
not be included in the list.
- All extracted claims should be standalone that can be understandable and verifiable without
additional context.
- You should preserve the original wording where possible, but provide necessary context to
make the claim self-contained. Particularly, use the context to recover pronouns, anaphoric
references (e.g. \"the paper\", \"the idea\"), and other such information to make the claim
self-contained. Use the name of entities rather than anaphors whenever possible.
- Along with the claims, you should also extract the corresponding citation URL(s) if they
exist. Citations can be in different formats:
- A segment of text + [number], for example: \"Li Qiang constructed a socioeconomic status
index (SES) based on income, education, and occupation, dividing society into 7 levels [15]\"
- A segment of text + [number†(some line numbers, etc.)], for example: \"Li Qiang constructed a
socioeconomic status index (SES) based on income, education, and occupation, dividing society
into 7 levels [15†L10][5L23][7†summary]\"
- [Citation Source](Citation Link), for example: \"Bolsonaro's rhetoric and frequent
conflicting signals (e.g. encouraging gatherings) eroded public trust in institutions [pmc.nc
bi.nlm.nih.gov](https://pmc.ncbi.nlm.nih.gov/articles/PMC11042250/#:~:text=Conclusion).\"
If the citation format is among the first two, please refer to the references/sources section
at the end to find the corresponding URLs for each claim.
- If a claim has no corresponding citation to support it, return an empty list for the url
field.
"- If multiple claims are associated with the same citation, extract them as separate entries.
If a claim has multiple citations, include all citation URLs in the url list.


Output format:
Return a list of JSON objects of the following format: [{\"claim\": \"EXTRACTED CLAIM TEXT\",
\"url\": [\"URL1\", \"URL2\", ...]}, ...].
Output only the JSON list directly, without any chitchat or explanations. If the highlighted
section does not contain any verifiable factual claims, please return an empty list. Please
make sure the URLs are copied verbatim from the original citations. The \"url\" field should be
a emtpy, single-item, or multi-item list.
```

Figure 16: **Claim Extraction System Prompt.**

```
Claim Extraction User Prompt

Extract the verifiable factual claims from the highlighted section of the report.\n
<report>\n{report}\n</report>\n
<highlighted_section>\n{highlighted_section}\n</highlighted_section>
```

Figure 17: **Claim Extraction User Prompt.**

```
Supported Judge System Prompt

You will be provided with a reference content (in <reference_content></reference_content> tags)
and a claim or statement (in <claim></claim> tags). Your task is to determine whether the claim
is 'supported', 'insufficient', or 'contradictory' with respect to the reference. Please note:

- 'supported': the claim is clearly supported by the reference.
- 'insufficient': the claim is weakly supported by the reference, or the reference is missing
key evidence, or the claim is not related to the reference.
- 'contradictory': the claim contradicts the reference.
First, assess whether the reference contains any valid content. If the reference contains no
valid information, such as a 'page not found' message, then the claim should be considered
'insufficient'. Then, carefully read the reference and the claim, and determine the
relationship between the claim and the reference. The reference content can be from one or
multiple webpages.

Output Format: Return a JSON string with the following format: {\"result\": \"supported\" |
\"insufficient\" | \"contradictory\"}. Do not include any other text or comments in your
response. Please make sure the result is based purely on whether the claim is supported by the
reference, not any other factors.
```

Figure 18: **Supported Judge System Prompt**

```
Supported Judge User Prompt

Judge if the cited reference content supports the claim.\n
<reference_content>\n{url_content}\n</reference_content>\n
<claim>\n{claim}\n</claim>
```

Figure 19: **Supported Judge User Prompt.**

Content Summarization System Prompt

```
You are a webpage summarization assistant. Your goal is to create a summary that preserves the
most important information from the original web page. Given scraped webpage in markdown format
(in <webpage_content></webpage_content> tags) and a list of claims (in <claims></claims> tags),
extract and summarize the parts of the webpage that are relevant to the claims.

Make sure you include all information that could support, contradict, or provide context for
the claims. Also, preserve as much other key information in the webpage as possible to provide
comprehensive context and be self-contained.

Try to use the original wording of the webpage content as much as possible.
If you find the webpage content is irrelevant to the claims, just generally summarize the web
page content covering all key information.
When you are summarizing, DO NOT use the third-person perspective (e.g. the webpage states that
..., the author says that ..., etc.). Just consider you are shortening the webpage as the
author.

Be as objective as possible and do not make any judgement or comments on the content.

Aim for about 20 percent of the original length, unless the webpage is already concise.
```

Figure 20: **Content Summarization System Prompt**

Content Summarization User Prompt.

```
Summarize the webpage content that are potentially relevant to the claims.\n
<webpage_content>\n{content}\n</webpage_content>\n
<claims>\n{claims}\n</claims>\n
Provide a summary of the webpage content. Preserve the original wording of the webpage content
as much as possible, and include all meaningful details. Do not include any other text or
explanations in your response.
```

Figure 21: **Content Summarization User Prompt.**

Rubric Evaluation System Prompt.

```
You will be given a question the user asked (in <question></question> tags) and the
corresponding report (in <report></report> tags) given as a response to the question by an
assistant. You will then be given a specific criterion to evaluate the report against (in
<criterion></criterion> tags).

Your task is to score the report based on whether it satisfies the criterion or not: 1 if the
report satisfies the criterion and 0 if the report does not satisfy the criterion. You might
also be asked to give score=-1 when the criterion is not applicable to the report. Please do
that when instructed. Judge only the specified aspect(s) in the criterion, not any other
qualities of the report. Please also provide a short (2 sentences maximum) justification for
your score.

Output only a JSON string with the following format: {\"score\": int, \"justification\":
string}. Do not include any other text or comments in your response.
```

Figure 22: **Rubric Evaluation System Prompt.**

Rubric Evaluation User Prompt

Evaluate the report based on the given criterion. If the criterion is not applicable to the report, score should be -1 instead of 0/1.
<question>\n{question}\n</question>\n
<report>\n{report}\n</report>\n
<criterion>\n{criterion}\n</criterion>

Figure 23: **Rubric Evaluation User Prompt.**

Pairwise Judge System Prompt

You will be given a research question the user asked (in <question></question> tags) and two versions of the report, original (in <report></report> tags) and revised (in <revised_report></revised_report> tags) that are generated by an assistant. The revised report is a revised version of the original report based on the feedback (in <feedback></feedback> tags) provided by the user.

Your task is to score the revised report based on whether it incorporates the feedback provided by the user or not, comparing it to the original report: 1.0 if the revised report incorporates the feedback, 0.5 if the revised report partially incorporates the feedback, and 0.0 if the revised report does not incorporate the feedback.

Output only a JSON string with the following format: {\"score\": float}. Do not include any other text or comments in your response. Please make sure the score is based purely on whether the feedback is reflected in the revised report compared to the original report, not any other factors.

Figure 24: **Pairwise Judge System Prompt.**

Pairwise Judge User Prompt

Score the revised report based on whether it incorporates the feedback provided by the user compared to the original report.
<question>\n{question}\n</question>\n
<report>\n{report}\n</report>\n"
<revised_report>\n{revised_report}\n</revised_report>\n
<feedback>\n{feedback}\n</feedback>

Figure 25: **Pairwise Judge User Prompt.**

Prompt Engineering System Prompt

```
You are an expert technical editor. Your task is to translate high-level user feedback into a
minimal, localized edit plan.

Input You Will Receive:
1) Original Query
2) Full Research Report
3) Original Feedback (often vague or high-level)

Your Goal:
Create a structured **edit plan** that enables a research agent to take specific, localized
editing actions without ambiguity.
Do NOT write or fabricate final factual content — your job is to **identify where** and **what
type** of content needs to be added/changed.

---

Editing Constraints:

- Only use the following atomic actions: **DELETE / INSERT / MODIFY**
- Every edit must specify an exact location with:
  - `Section` name (must match exactly)
  - `Subsection` name (or N/A if not applicable)
  - `Anchor` quote: A short (18 words) **verbatim** sentence/phrase from the current report
  that clearly identifies **where** the edit should occur.
- Reference the Anchor in your **Content Spec** to clarify where in the text the change happens.
- INSERT actions can create new sections/subsections, but only if explicitly specified in the
feedback.
- Do NOT invent specific facts (names, numbers, dates, benchmarks, claims).

---

Output Format (Markdown):

Feedback:
[Insert original feedback exactly as received]

Edit Actions:
1) Action: DELETE | INSERT | MODIFY
   Location:
   - Section: "[Exact section name]"
     *(For new sections, use format: `NEW: [Section Name]`)*
   - Subsection: "[Exact subsection name]" (or N/A)
     *(For new subsections, use format: `NEW: [Subsection Name]`)*
   - Anchor: "[Short verbatim quote from report]"
     *(For new sections/subsections, specify relative location, e.g., "After section
     'Discussion')*
   Content Spec:
   - What to change: Describe required content, not final prose
   - Must-include: Specific elements that must be part of the edit
```

Figure 26: **Prompt Engineering System Prompt.**

Prompt Engineering Hard-coded Constraint Suffix

```
You are given:
1) The original user feedback
2) A structured list of localized Edit Actions derived from that feedback

Each Edit Action includes:
- Action: One of DELETE, INSERT, or MODIFY — the atomic type of edit to apply.
- Section / Subsection: The precise location in the document where the edit applies. If the
action introduces a new section/subsection, it will be labeled as `NEW: [Name]`.
- Anchor: A short verbatim quote from the report identifying the exact insertion/modification
point. For new sections/subsections, this is a relative reference (e.g., "After section
'Discussion'").
- Content Spec: A short explanation of what to change, localized to the Anchor location. This
is NOT final content — only a structural and intent-level guide.

Non-negotiable editing constraints:
- Apply ONLY the actions listed under "Edit Actions".
- Do NOT infer, add, or modify edits beyond what is explicitly specified.
- Do NOT reinterpret or expand the original feedback.
- Do NOT rewrite sections wholesale; keep edits strictly local to the specified Anchor quote.

Your Task:
Apply the Edit Actions to improve the report by making precise, localized edits at the
specified locations, adhering strictly to all constraints above. Please only output the
revised report and no other text such as comments or explanations.
```

Figure 27: **Prompt Engineering Hard-coded Constraint Suffix.**

Reviser Subagent System Prompt

```
You are a research report revision assistant. Your task is to revise and improve a research
report based on user feedback.

You have access to a web search tool to find additional information, verify facts, or gather
supporting evidence. Note that you can only use the web search tool for {max_tool_calls} times.
If you have used the web search tool for {max_tool_calls} times, you should then produce the
final report and stop using the web search tool again.

## Guidelines

1. **Understand the feedback**: Carefully read the user's feedback to understand what needs to
be improved.
2. **Search when needed**: If the user feedback requires searching for additional information,
use the web search tool to find the information.
3. **Maintain Quality**: Ensure the revised report locally addresses the user feedback without
making any changes to other parts.
```

Figure 28: **Reviser Subagent System Prompt.**

Reviser Subagent User Prompt

```
## Original Research Question

{question}

## Current Report

{report}

## Feedback for Revision

{feedback}

---

Please revise the report based on the feedback above. Use web search if you need additional
information. Return ONLY the revised report and no other text such as comments or explanations.
```

Figure 29: **Reviser Subagent User Prompt.**