SciELF: A Benchmark for Long-form Scientific Factuality Assessment

Anonymous ACL submission

Abstract

The verification of long-form scientific responses remains a crucial yet under-explored challenge in natural language processing. Existing verification benchmarks predominantly evaluate isolated, short-form claims, failing to address the difficulty of localising factual errors within complex, multi-claim narratives. To address this, we present **SciELF**, a benchmark for evaluating fact-checking systems on long-form scientific texts. Built from Open Access articles, SciELF features expert-authored long-form answers that are systematically corrupted to introduce realistic factual errors. Furthermore, to move beyond broad document-level judgments, we decompose these long-form texts into atomic claims. Each atomic claim is independently annotated for ground-truth factuality, enabling fine-grained evaluation of a system's ability to identify errors within long-form text. We scale the benchmark by supplementing 100 human-curated examples with 868 synthetically-generated, expert-verified answers. Evaluating three recent fact-checking systems in both closed (source-article evidence) and open (web-retrieved evidence) settings, we find that while systems perform reasonably well in closed settings (F1 = 0.867), performance degrades significantly in open settings (F1 = 0.731) where appropriate evidence is harder to isolate. Finally, we conduct an extensive error analysis identifying distinct failure patterns and open challenges in reliable scientific factuality assessment.

1 Introduction

Long-form scientific response verification represents a challenging and under-studied problem in natural language processing. The ability to verify whether a scientific response is factually consistent with established knowledge is increasingly important as large language models (LLMs) are deployed in knowledge-intensive settings such as automated literature discovery (Eger et al., 2025; Ramachandran et al., 2026) and research synthesis (Asai et al., 2026) workflows. The majority of existing work in long-form factuality focuses on general domains, such as biographies, historical texts, or online content (Min et al., 2023; Song et al., 2024). These approaches are typically evaluated on Wikipedia-derived content, where factual claims are relatively self-contained and evidence is readily retrievable. While there exist factuality benchmarks in the scientific domain (Wadden et al., 2020), they primarily focus on *short-form* claims, where the evaluated text consists of a single sentence or isolated fact. In contrast, long-form scientific responses contain numerous interdependent claims that incorporate background knowledge, experimental findings and results, and domain-specific reasoning.

To address factuality verification, recent work has proposed automatic fact-checking systems (AFCs) that verify text by first decomposing it into atomic statements or claims, retrieving relevant evidence for each claim, and using the retrieved evidence to determine whether the claim is supported or not (Min et al., 2023; Wei et al., 2024b). Systems such as FActScore (Min et al., 2023), VERIFY (Fatahi Bayat et al., 2025), and FactReasoner (Marinescu et al., 2025) follow this protocol and are the AFCs we consider in this study. As automated research assistants are becoming ubiquitous, the need for rigorous evaluation of their outputs is of increasing importance. Naturally, we can think of applying existing AFCs on the outputs of these tools. Although existing works have demonstrated the reliability of AFCs in verifying claims in general domains such as on Wikipedia and biography content, their performance on text from the scientific domain remains to be studied.

In light of these observations, we propose **SciELF** (**Sci**entific **E**valuation of **L**ong-form **F**actuality), a benchmark for evaluating AFCs in the scientific domain. SciELF is built from Open

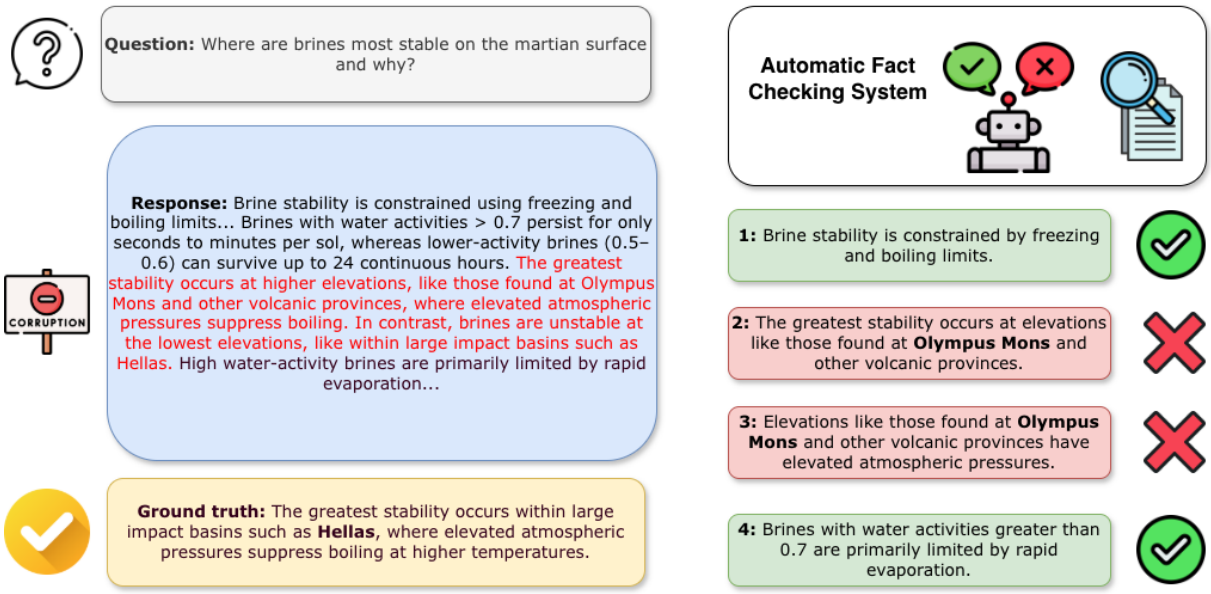

Figure 1: An example from SciELF. Here a response is corrupted (red span) while some of the original text is preserved. The atoms corresponding to original spans should be supported, while the atoms corresponding to falsified spans are not-supported. The ground truth shows the correct span from the original answer.

Access scientific articles in which Subject Matter Experts (SMEs) curate a question and a comprehensive answer based on an article's content. These answers are then systematically corrupted by the SMEs to falsify specific information, after which they are decomposed into atomic claims and annotated for ground-truth factuality. To scale dataset size, the human-curated portion is supplemented with synthetically-generated question-answer pairs that are validated by the SMEs before inclusion. By deriving all content exclusively from Open Access articles, SciELF facilitates fully reproducible research. Furthermore, the mixture of factual and non-factual claims requires systems to reliably discriminate between the two, while atom-level ground-truth annotations provide fine-grained insights into system predictions.

Figure 1 illustrates an example from SciELF. Given the question *"Where are brines most stable on the Martian surface and why?"*, a long-form response is systematically corrupted by replacing the correct location, the low-elevation impact basin Hellas Planitia, with the high-elevation volcanic plain Olympus Mons. This substitution concerns the relationship between elevation and atmospheric pressure on Mars: at low elevations such as Hellas, higher atmospheric pressure raises the boiling point of brines (suppressing evaporation), allowing them to persist as liquids for longer. At high elevations such as Olympus Mons, lower atmospheric pressure induces rapid evaporation and boiling, making brines far less stable. The corrupted response thus inverts the true geophysical relationship. In this example, two listed atoms derived from the corrupted span are correctly identified as non-supported, while the remaining claims are supported. Identifying these errors requires successful evidence retrieval and strong reasoning capabilities about scientific phenomena, a demand that is arguably more challenging than biographical content.

Contributions Our main contributions are: (1) We present **SciELF**, a benchmark containing 100 expertly-curated and 868 synthetically-generated long-form answers, decomposed into atomic claims and annotated for ground-truth factuality.[1] (2) We evaluate three AFCs on SciELF in both closed and open retrieval settings, providing the first systematic comparison of such systems on scientific domain text. (3) We conduct an extensive error analysis that provides insights into the failure patterns of AFCs on scientific content.

2 Related Work

Initial work on factuality includes the FEVER benchmark (Thorne et al., 2018), where claims are automatically corrupted to introduce false information, while Maynez et al. (2020) analysed

[1] Our dataset will be made available upon publication under a CC-BY-NC-4.0 license, an anonymised version is listed here: https://anonymous.4open.science/r/SciELF-1A70/

the level of hallucinations produced by language models in summarisation tasks. Subsequent research has expanded on these themes by developing specialised benchmarks. Short-form benchmarks such as TruthfulQA (Lin et al., 2022) and SimpleQA (Wei et al., 2024a; Haas et al., 2025) focus on single-answer factual queries. In contrast, long-form benchmarks like FActScore (Min et al., 2023) and LongFact (Wei et al., 2024b) evaluate paragraph-level responses against Wikipedia, biographies and diverse web sources, respectively.

A common approach in long-form factuality evaluation is to decompose a response into *atomic claims*: short, self-contained statements each conveying a single piece of information (Min et al., 2023; Wei et al., 2024b; Gunjal and Durrett, 2024). Systems following this atomise-retrieve-verify paradigm primarily differ in how they use retrieved context. For instance in FActScore, each atom is used to formulate a search query, and the results are used to directly support or refute the claim, while in VERIFY (Fatahi Bayat et al., 2025) they use an iterative query refinement approach on retrieved web content, adjusting the search query in a multi-round setting to extract the most relevant evidence. Subsequent work such as FactReasoner (Marinescu et al., 2025) has developed a more complex verification procedure: a Probabilistic Graphical Model is built that captures entailment and contradiction relationships between atomic statements and evidence sources. Inference methods are then run on the graph to calculate the posterior probability of each atom being supported or not.

While the previous-mentioned works concern factuality evaluation, their scope is limited to the general domain. Within the scientific domain, foundational efforts like SciFact (Wadden et al., 2020) solely focus on short-form claim verification, where abstracts are used to support or refute claims from papers. To address the need for deeper comprehension, benchmarks such as QASA (Lee et al., 2023) have been created that require "full-stack reasoning" to provide appropriate evidence-based rationales in answers to scientific questions. However, while QASA evaluates a model's ability to generate answers to complex questions, it does not provide a mechanism to systematically verify the truthfulness of the resulting long-form text.

Recently, systems have been developed to perform automated research synthesis (Eger et al., 2025). Frameworks like OpenScholar (Asai et al., 2026) evaluate multi-document generation using high-level rubrics such as coverage, relevance, and citation accuracy, but lack explicit checks on underlying factuality. We have created SCIELF to meet the need for a benchmark that can evaluate how systems perform at claim-level veracity predictions in long-form scientific text.

3 The SciELF Dataset

3.1 Manual Example Creation

For the human-curated portion of SCIELF, five SMEs with expertise spanning diverse scientific disciplines, including biomass estimation, astrophysics, and meteorology were tasked with collating a list of papers in their area of study that they have familiarity with. The SMEs then created a number of questions based on the content of a paper, and for each question, wrote a long-form answer. The answer is referred to as a *ground truth answer*. Then, for each ground truth answer, the SME was asked to *corrupt* information such that the answer would contain false information. This type of answer is referred to as a *corrupted answer*. The types of corruption involved: *context corruption*, where information grounded in the paper is distorted or misrepresented; *external knowledge corruption*, where external claims or background information are introduced; *numeric alteration*, where quantitative details such as counts, percentages, or statistical values are modified; and *reasoning distortion*, where the logical relationship between evidence and conclusions is altered while preserving some surface-level facts.

3.2 Synthetic Example Creation

The manual curation of dataset examples does not scale easily due to the time required to author and corrupt long-form answers. To increase dataset coverage while maintaining the same structure and corruption categories, we introduce a semi-automated pipeline for generating synthetic question-answer examples from the same scientific papers as in the manual portion. The goal of this pipeline is to mirror the manual procedure described above: generate a question grounded in the paper, produce a correct long-form answer (corresponding to the ground truth answer), and then construct a corrupted version of that answer following the same corruption taxonomy as in Section 3.1.

Each PDF is processed through a sequence of stages: text extraction, semantic segmentation, hierarchical summarisation, and schema-

constrained QA generation. First, the document text is extracted and divided into paragraphs, which serve as the basic semantic units. Paragraph embeddings are then computed using the `text-embedding-3-small` model, allowing semantically related paragraphs to be grouped into coherent chunks. Chunk boundaries are determined using both length constraints and cosine similarity between adjacent paragraph embeddings. Specifically, a new chunk is created when either the current chunk exceeds a maximum character limit (5,000 characters) or when the chunk has reached a minimum size (1,500 characters) and semantic similarity drops below a threshold (0.75). This approach preserves topical coherence while keeping chunks small enough for reliable downstream processing.

Each chunk is summarised independently using an LLM, with outputs constrained by JSON schema validation to ensure consistent structure. These chunk summaries are then aggregated into a structured representation of the entire paper using another schema-constrained LLM call. The resulting global summary captures key elements of the paper, including the research question, background, methods, results, discussion, limitations, and conclusions. This structured representation serves as the primary substrate for synthetic QA generation.

From this global representation and the supporting document content, the system generates question-answer examples. Each example contains a question, a concise factual answer, a long-form comprehensive answer grounded in the paper, and a falsified or corrupted passage designed to resemble plausible scientific text while containing incorrect information. Additional fields capture the type of corruption applied, any domain knowledge required to interpret the example, and whether the generated text includes citation-like references. For each paper, the pipeline generates one QA example corresponding to each corruption type.

Although examples are generated automatically, all synthetic instances undergo human validation before inclusion in the final dataset. The same SMEs who authored the manual examples review the synthetic ones using a custom web application that displays the question, generated answers, corruption type, and associated paper context. SMEs can approve an example as-is, edit it to correct or improve it (an important functionality given that there can be one false claim in otherwise truthful text), or reject it entirely. This review stage ensures that synthetic examples remain scientifically plausible, adhere to the corruption definitions used in the dataset, and maintain the overall quality required for reliable evaluation. An instance of an original and corrupted synthetic example is given in Table 6 in Appendix E.

3.3 Annotation Pipeline

The next stage in the creation of SCIELF involves fine-grained annotation of the claims present in the answers. To accommodate this, we built a FastAPI[2] application to generate annotation files. We built an assessor pipeline based on FactReasoner (Marinescu et al., 2025) that operates on the input question and answer. We chose this pipeline as it contains modular components for text *atomisation* and *decontextualisation*, evidence *retrieval*, and *inference* (predicting the label of each atom). Note that the retrieval and inference components can be swapped out to enable other systems, e.g. FActScore (Min et al., 2023) and VERIFY (Fatahi Bayat et al., 2025), which differ in terms of the prompting strategies used, and how the evidence is inferenced upon.

The long-form answer is split into self-contained atomic units that are annotated for ground-truth factuality by a human annotator. The annotator is provided with the original ground-truth answer, the corrupted answer, and the specific text spans that were corrupted. Annotating individual atoms involves mapping the atom back to its relevant sentence in the long-form text. Atoms corresponding to the original, ground-truth spans are labeled as True, whereas atoms corresponding to the corrupted spans are labeled as False. Note that a corrupted answer can still contain spans of text that are considered true. This yields a definitive binary (True/False) annotation for each atom, allowing for the fine-grained calculation of metrics such as Accuracy, Precision, Recall, and F1-score for AFCs.

3.4 Automatic Annotation of Synthetic Examples

While all atoms for the manually-created portion of the dataset were annotated for ground truth factuality by a human annotator (Section 3.3), this process becomes unfeasible when considering the large number of synthetically-generated atoms (11,101). Since annotating the label of an atom is largely a problem of deducing whether the atom corresponds

[2] https://github.com/fastapi/fastapi

to a corrupted sentence, we experimented with an LLM-based automatic annotator that assigns "silver" truth labels to the atoms. For each corrupted answer, the LLM-based labeller is provided with the question, the ground-truth factual answer, the corrupted passage, the corruption type, and the description of the corruption. In order to test whether this may be enough evidence for the model to reliably predict the atom's label, we ran the labeler on the manually-created and human-annotated atoms as a development set. We also considered a setting where we add the text of the full PDF alongside the above information.

Table 1 reports the results of this auto-annotation experiment using GPT-4.1 as the labeler. Using the metadata and appropriate spans only, we achieved 0.891 labeling F1. Adding the full PDF substantially improves performance with respect to the metadata-only condition (0.946 vs. 0.891). For 50/434 instances (11% of corrupted atoms), the SMEs annotated the synthetic atoms for ground truth factuality. Agreement rates between the automatically predicted labels and their assessments ranged from 76.9% to 93.3% among annotators. At F1 = 0.946 and relatively high agreement between the automatic and SME annotations, we consider these silver labels sufficiently reliable to use in the synthetic portion of SCIELF, especially for highlighting failure modes and then applying further human scrutiny, though fine-grained system comparisons should prioritise the human-curated split.

Table 1: Auto-annotation with GPT-4.1 as the labeler on the human-annotated corrupted split (819 atoms).

Condition	Precision	Recall	F1
Metadata only	0.898	0.884	0.891
+ Full PDF	0.948	0.944	**0.946**

3.5 Dataset Statistics

The manually created portion of the dataset contains 28 papers, with 50 factual and 50 non-factual answers. The breakdown of the number of atoms is given in Table 2. In total there are 1,667 atoms or claims across all 100 manual answers. Of these, 1,417 atoms (85.0%) are true, and 250 atoms (15.0%) are false. For the synthetically-generated portion of the dataset, there are 434 factual and 434 non-factual answers. Across the full dataset of 968 answers, there are 12,768 atoms in total, of which 10,437 (81.7%) are true and 2,331 (18.3%) are false. The higher number of positive class labels ensures a setting where AFCs must be able to reliably distinguish erroneous claims among largely true text.

Table 2: Atom-level dataset statistics in SciELF. Gray cells denote silver labels assigned by the GPT-4.1 auto-annotator (F1 = 0.946 on the human-annotated split).

Statistic	Orig.	Corr.	Comb.
Manual			
Answer instances	50	50	100
Total atoms	848	819	1,667
True atoms	848	569	1,417
False atoms	0	250	250
Atoms/inst. (mean)	17.0	16.4	16.7
Synthetic			
Answer instances	434	434	868
Total atoms	5,819	5,282	11,101
True atoms	5,819	3,201	9,020
False atoms	0	2,081	2,081
Atoms/inst. (mean)	13.4	12.2	12.8
Overall			
Answer instances	484	484	968
Total atoms	6,667	6,101	12,768
True atoms	6,667	3,770	10,437
False atoms	0	2,331	2,331
Atoms/inst. (mean)	13.8	12.6	13.2

4 Experiments

We evaluate recent AFCs on the SCIELF benchmark across two retrieval environments (closed and open). This allows us to assess baseline reasoning ability when gold standard evidence is available, and analyse how system behaviour changes when faced with the more diverse and noisy evidence retrieved in the open setting.

Preliminaries Let r denote a long-form response either written by an SME or generated by a language model to an input question x. We decompose r into a set of atomic claims using a dedicated extraction model:

$$\mathcal{A} = \{a_1, \ldots, a_n\} = LLM_{\text{atom}}(r) \quad (1)$$

Each atom a_i is a minimal, independently verifiable unit of information. Initially, the extracted atoms may contain implicit references (e.g. pronouns or ellipsis), we apply a revision step that rewrites each atom into a fully self-contained statement:

$$a_i^* = LLM_{\text{revise}}(a_i, r) \quad (2)$$

For each revised atom a_i^*, we retrieve a set of supporting contexts $\mathcal{C}_i = \{c_1, \ldots, c_k\}$ from an evidence source (either the source document or the

web), where k is the number of retrieved passages. $\mathcal{C} = \bigcup_{i=1}^{n} \mathcal{C}_i$ is the full evidence pool across all n atoms.

A fact-checking component then assigns a binary label $y_i \in \{\text{S}, \text{NS}\}$ to each atom, indicating whether it is *Supported* or *Not Supported* by its associated contexts[3]:

$$y_i = \textit{NLI}(a_i^*, \mathcal{C}_i) \tag{3}$$

Let $l_i \in \{\text{S}, \text{NS}\}$ denote the ground-truth label for atom a_i^*, and let $\mathcal{L} = \{l_1, \ldots, l_n\}$ and $\mathcal{Y} = \{y_1, \ldots, y_n\}$ denote the full sets of ground-truth labels and system predictions, respectively. Performance is measured by comparing $\mathcal{Y}$ against $\mathcal{L}$ using standard classification metrics (Accuracy, Precision, Recall, and F1).

Closed Setting For the first series of experiments, we consider a closed setting, where evidence is retrieved from the individual article. The single PDF article content is extracted with Docling (Team, 2024), and the corresponding text chunks are indexed into a vector database.[4] We retrieve the top $k = 3$ chunks per atom $a_i^* \in \mathcal{A}$ using the `all-MiniLM-L6-v2` encoding model, forming the context set $\mathcal{C}$ used for fact-checking.

Open Setting For the next set of experiments, we consider an open setting where evidence is retrieved via web search. We reuse the revised atoms $\mathcal{A}^* = \{a_1^*, \ldots, a_n^*\}$ from the closed setting but replace the context sets $\mathcal{C}_i$ with passages retrieved via web search. For web search, we use the Serper API.[5] Here the atom text is reformulated as a search query by an LLM, and we use the Google search engine to retrieve the first 10 items. If the item is a web page, we extract it with the `HTML2text` library, or if the item is a PDF article, it is extracted with `PDF2Text`. If a PDF article is behind a paywall or is copyright-restricted, we remove the item from the set of candidates and process the next valid item. The extraction process is finished when we have collected $k = 5$ extracted pages. We use a larger value for k to compensate for the fact that web-based retrieval can include noisier examples. In order to retrieve the most important information from each of the k items, the item is split into text chunks and indexed into a temporary vector store, from which the top 3 chunks are returned.

4.1 Fact-checking Systems

Here we describe the fact-checking systems used in our evaluation. We follow the approach of recent fact-checking systems that atomise long-form answers into isolated, self-contained units of information (Min et al., 2023; Fatahi Bayat et al., 2025; Marinescu et al., 2025), and predict a factuality label for each extracted atomic statement. For all systems in this setting we use the Llama-3.3-70b-Instruct model (Touvron et al., 2023) as the LLM, which is an open-source model that strikes a balance between performance and computational cost.

All atoms $\mathcal{A}$ are shared across fact-checking systems, and the systems differ in their prediction component by utilising different prompting strategies and how they model evidence between the atoms and contexts. We follow (Marinescu et al., 2025) who summarise the retrieved contexts using an LLM:

$$c_i^* = \textit{LLM}_{\text{summarise}}(c_i, a_i) \tag{4}$$

In the case where a context is deemed to be not relevant to an atom by the LLM, the context is removed. This step is carried out so that we only give credit to a system that uses valid evidence to make its decision (Thorne et al., 2018; Schlichtkrull et al., 2023).

FActScore The first system we consider is FActScore (Min et al., 2023), particularly their *Retrieve LM* setting that achieved the highest-performing micro F1 score in their experiments.[6] This represents a baseline that uses localised contexts $\mathcal{C}_i$ (contexts directly retrieved for an atom) to predict the factuality label of each atom.

VERIFY Next we consider VERIFY (Fatahi Bayat et al., 2025), which uses Chain-of-Thought reasoning to predict the label of each atom. Like FActScore, VERIFY only considers local contexts when predicting an atom's label. For retrieval, VERIFY utilises an n-round interactive procedure: given an atom, a query is generated and is used to retrieve items from a web search. We extract the first k items, and use the same temporary vector-store

[3] Other systems can assign an *unverifiable* or *not-enough-evidence* label.

[4] In our pipeline we use ChromaDB (`https://github.com/chroma-core/chroma`)

[5] `https://serper.dev/`

[6] We do not consider their *Non-Parametric* setting as it involves running a Masked Language Model which requires access to a dedicated GPU and is thus less suitable for production environments.

based retrieval to get the most relevant chunks per item. The text contents are concatenated and then passed to the summariser LLM to make it appropriate for the atom. The summary is passed back to the query builder, which refines the query for the next round, adapting the query to retrieve more relevant information. This process repeats for n rounds, yielding up to n per-round summaries that serve as the evidence passages for the atom verifier.

FactReasoner As our final fact-checking system we consider FactReasoner (Marinescu et al., 2025). FactReasoner approaches the verification process using a probabilistic graphical model (PGM), that models relationships between atoms and *all* retrieved contexts, before running a graphical inference algorithm to compute the posterior probabilites of each atom being Supported/Not Supported. By reasoning over the entire set of contexts $\mathcal{C}$ simultaneously, the system can weigh supporting and conflicting signals. We reuse the retrieved contexts contexts $\mathcal{C}$ from the FActScore setting.

5 Results and Analysis

5.1 Comparison of Fact-checking Systems on SciELF

Table 3 reports atom-level performance of FactReasoner, FActScore, and VERIFY across in-domain and web-based retrieval for the manually-created portion of the dataset. All metrics are macro-averaged over individual answer instances. We report the corrupted and combined splits. Results on the synthetic portion of the dataset are listed in Appendix C.

FactReasoner has the highest recall among all systems at 0.828 on the combined data, and achieves the highest F1 score of 0.867, which is .07 points higher than the next best system FActScore. VERIFY achieves the highest precision at 0.958. FactReasoner's considerable advantage in terms of recall stems from its ability to incorporate contexts retrieved from other atoms into the prediction of a given atom, which is not the case with FActScore or VERIFY.

In the web-based setting, performance drops across all systems (e.g., from 0.867 to 0.731 in the case of FactReasoner), reflecting the additional challenge of verifying highly-specific claims with web-retrieved evidence, where the retrieved evidence may not include the source paper. Despite VERIFY's more complex interactive query-refinement and retrieval approach, it still underperformed the other two systems with a combined F1 score of just 0.579.

Table 3: Atom-level performance on the manually-created portion of the data. *In-Domain* uses passages from the source paper; and *Web* uses web-retrieval; *Corr* refers to the corrupted portion and *Comb* the combined original and corrupted portions.

System	Split	Acc.	Prec.	Rec.	F1
In-Domain					
FactReasoner	Corr.	**0.819**	0.890	**0.837**	**0.841**
	Comb.	**0.819**	0.945	**0.828**	**0.867**
FActScore	Corr.	0.778	0.890	0.724	0.761
	Comb.	0.746	0.945	0.719	0.790
VERIFY	Corr.	0.759	**0.916**	0.673	0.724
	Comb.	0.728	**0.958**	0.685	0.766
Web					
FactReasoner	Corr.	**0.703**	0.803	**0.698**	**0.694**
	Comb.	**0.673**	0.902	**0.671**	**0.731**
FActScore	Corr.	0.591	0.774	0.527	0.565
	Comb.	0.536	0.887	0.504	0.599
VERIFY	Corr.	0.589	**0.810**	0.481	0.588
	Comb.	0.505	**0.905**	0.451	0.579

5.2 Analysis of SCIELF

We report high-level findings from the results of FactReasoner on the manually-created portion of the data. Detailed breakdowns and case studies of particular error types are in Appendix D.

Graph structure improves recall FactReasoner's ability to model inter-atom context relationships leads to its considerable recall advantage over FActScore (0.828 vs. 0.719). We identify 76 atoms that did not retrieve any relevant contexts yet were correctly predicted as supported by relying on passages received for other atoms in the graph. FActScore, which retrieves evidence and verifies each atom in isolation, suffers when no appropriate information was retrieved for an atom.

Error rates by atom type and corruption category In order to gain insight into what type of atoms a system struggles to predict, we classified atoms into the following categories using GPT-4o-mini: *citation* (references a specific author, year, or publication), *numerical* (contains specific quantities, statistics, or measured values), *methodological* (describes a procedure, technique, or experimental design), *comparative* (explicitly compares two entities, conditions, or values), *causal* (uses inferential language such as "suggests," "indicates," or "leads

Table 4: Error rate by atom category. The *Example atom* column shows a representative atom from that category.

Category	Example atom	Errors	Correct	Error rate
Citation	"*Cowie et al. published an optimized averaging method in 2012.*"	136	246	35.6%
Numerical	"*Hyperparameter tuning improved* R^2 *from* 0.19 *to* 0.60."	113	296	27.6%
Methodological	"*The quality control discarded pixels with* > 4 *shifts during the 1972–2017 period.*"	59	185	24.2%
Claim	"*Faster-moving tropical cyclones have shear-aligned convection.*"	131	571	18.7%
Comparative	"*UCM tended to displace rainfall westward for Hurricane Florence.*"	35	168	17.2%
Causal	"*The enhanced rainfall in Houston is due to stronger building drag.*"	115	766	13.1%
Negation	"*Faster storm translation does not merely redistribute rainfall in time.*"	10	72	12.2%

to"), *negation* (explicitly negates a relationship or states an absence or limitation), and *claim* (a general factual statement fitting none of the above).

Table 4 reports the error rate (FP + FN / total atoms in that category) for each category. Citation atoms pose the greatest challenge (35.6% error rate). These atoms often reference a specific author and year of publication, and retrieval systems that rely on semantic similarity often fail to retrieve these passages. Furthermore, even if citation metadata is successfully retrieved from the references section, it is often filtered out by the summariser component for not containing enough substantive information to support the claim. Numerical atoms (27.6%) encounter an issue where the original value is often highly-specific and the summariser component generalises the value, rendering it unverifiable. Methodological atoms (24.2%) represent a distinct challenge as they contain unique details relevant mainly to the paper in question, putting more emphasis on better retrieval quality and reasoning ability, as knowledge of these details is not covered in general pre-training corpora. Conversely, causal and negation atoms are the easiest to verify (13.1% and 12.2% error rates, respectively), as inferential and negated phrases can usually be detected by capable LLMs where these claims are often contradicted by the main narrative of the paper.

By isolating the corrupted atoms (TNs and FPs) and categorising them by **corruption type** (Section 3.1), we gain a clearer picture of how FactReasoner handles different adversarial patterns. Despite the high error rate among numerical atoms, corruptions resulting from pure numeric alterations are relatively easy for the model to catch, yielding a 10.8% miss rate across 12 instances. This corruption type usually preserves the surrounding context, and the different value is easily surfaced during retrieval. Surprisingly, reasoning distortions were accurately predicted for the most part, showing only an 11.0% miss rate across 14 instances. The system is resilient against these corruptions because inverted conclusions fundamentally contradict both the source paper's core narrative and established domain knowledge, allowing the verifier to easily detect them. On the other hand, combined numeric-plus-context corruptions present a more significant challenge (42.4% miss rate across 7 instances). By altering both a specific number and its surrounding text, this dual corruption creates a self-consistent false narrative. Because the fabricated numbers are combined with fabricated context, the retrieval becomes more noisy, as the query is semantically different to the original text.

6 Conclusion

In this work we introduce SCIELF, a benchmark for evaluating fact-checking systems on long-form scientific text, that comprises 100 human-annotated and 868 synthetic instances drawn from Open Access research articles. We evaluated three recent fact-checking systems under both in-domain and web-based retrieval settings. FactReasoner consistently outperformed the other systems across both settings, due to its ability to utilise evidence retrieved for one claim to support the verification of related claims within the same response. We release SCIELF alongside its evaluation pipelines as a testbed for the scientific factuality community, with the aim of spurring development of more robust, retrieval-augmented verification systems.

Limitations

While SciELF provides a rigorous testbed for evaluating automatic fact-checking systems on long-form scientific content, we note limitations that should be considered when interpreting benchmark results or deploying downstream systems. Firstly, due to the highly time-consuming nature of generating Question-Answer pairs and the corrupted versions, our manually created portion of the dataset is relatively small at 100 examples. Further, due to limited time commitments from the SMEs, we optimised for SME depth of experience and as a result, the papers selected for inclusion are limited to their domains of expertise, as opposed to a larger and more diverse collection of scientific corpora. This may introduce a sampling bias towards certain domains, and limits our applicability to wider scientific domains. We hope that greater depth of experience results in a more narrow set of examples that are constructed to a high level of quality, and our findings in these domains are applicable to wider domains which share some overlap (in terms of technical content, long-tailed domain, and the presence of numerical findings, reference to study parameters etc.).

A primary limitation of our work pertains to the synthetic split of the dataset. Although we mitigate LLM generation bias by requiring Subject Matter Experts to verify, edit, or remove GPT-4.1 generated claims, the atomic ground-truth annotations for these instances remain automatically derived (i.e., silver labels). Because automated prediction may struggle to correctly capture highly nuanced or long-tail scientific phenomena, we strongly encourage researchers to prioritise the purely human-curated test set when conducting fine-grained evaluations

Finally, the systems which we studied: FActScore, VERIFY and FactReasoner were used "off-the-shelf" and we did not perform customisations to adapt their prompts or components to the scientific domain. It is likely system performance can be improved by customising these components.

Ethical Considerations

This work introduces a novel benchmark dataset designed to assess automatic fact-checking systems on long-form scientific text, with the primary goal of identifying their specific success and failure modes. Ultimately, our aim is to enhance the trustworthiness and reliability of automated research tools by providing a clearer understanding of how fact-checking systems evaluate complex outputs. However, we caution that strong performance on our benchmark should not be interpreted as a definitive license to deploy automated research tools in sensitive, critical, or high-stakes scientific domains without expert human oversight. Furthermore, the factual corruptions introduced in this benchmark are strictly intended as a diagnostic tool for fact-checking systems. They should not be deliberately repurposed to deceive models or inject misinformation into downstream applications. Regarding human curated data and annotations, these contributions were conducted with informed consent by authors of this paper.

References

Akari Asai, Jacqueline He, Rulin Shao, Weijia Shi, Amanpreet Singh, Joseph Chee Chang, Kyle Lo, Luca Soldaini, Sergey Feldman, Mike D'Arcy, David Wadden, Matt Latzke, Jenna Sparks, Jena D. Hwang, Varsha Kishore, Minyang Tian, Pan Ji, Shengyan Liu, Hao Tong, and 9 others. 2026. Synthesizing scientific literature with retrieval-augmented language models. *Nature*, 650(8103):857–863.

Steffen Eger, Yong Cao, Jennifer D'Souza, Andreas Geiger, Christian Greisinger, Stephanie Gross, Yufang Hou, Brigitte Krenn, Anne Lauscher, Yizhi Li, Chenghua Lin, Nafise Sadat Moosavi, Wei Zhao, and Tristan Miller. 2025. Transforming science with large language models: A survey on ai-assisted scientific discovery, experimentation, content generation, and evaluation. *ArXiv*, abs/2502.05151.

Farima Fatahi Bayat, Lechen Zhang, Sheza Munir, and Lu Wang. 2025. FactBench: A dynamic benchmark for in-the-wild language model factuality evaluation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 33090–33110, Vienna, Austria. Association for Computational Linguistics.

Anisha Gunjal and Greg Durrett. 2024. Molecular facts: Desiderata for decontextualization in LLM fact verification. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 3751–3768, Miami, Florida, USA. Association for Computational Linguistics.

Lukas Haas, Gal Yona, Giovanni D'Antonio, Sasha Goldshtein, and Dipanjan Das. 2025. SimpleQA Verified: A reliable factuality benchmark to measure parametric knowledge. *CoRR*, abs/2509.07968.

Yoonjoo Lee, Kyungjae Lee, Sunghyun Park, Dasol Hwang, Jaehyeon Kim, Hong-in Lee, and Moontae Lee. 2023. Qasa: advanced question answering on

scientific articles. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland. Association for Computational Linguistics.

Radu Marinescu, Debarun Bhattacharjya, Junkyu Lee, Tigran T. Tchrakian, Javier Carnerero-Cano, Yufang Hou, Elizabeth M. Daly, and Alessandra Pascale. 2025. FactReasoner: A probabilistic approach to long-form factuality assessment for large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 14547–14577, Suzhou, China. Association for Computational Linguistics.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919, Online. Association for Computational Linguistics.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100, Singapore. Association for Computational Linguistics.

Rahul Ramachandran, Kaylin Bugbee, and Juan Bernabe-Moreno. 2026. Accelerated knowledge discovery: A vision for nasa science. *Earth and Space Science*, 13(3):e2025EA004662. E2025EA004662 2025EA004662.

Michael Schlichtkrull, Zhijiang Guo, and Andreas Vlachos. 2023. Averitec: a dataset for real-world claim verification with evidence from the web. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

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.

Deep Search Team. 2024. Docling technical report. Technical report.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana. Association for Computational Linguistics.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aur'elien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *ArXiv*, abs/2302.13971.

David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020. Fact or fiction: Verifying scientific claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7534–7550, Online. Association for Computational Linguistics.

Jason Wei, Nguyen Karina, Hyung Won Chung, Yunxin Joy Jiao, Spencer Papay, Amelia Glaese, John Schulman, and William Fedus. 2024a. Measuring short-form factuality in large language models. *CoRR*, abs/2411.04368.

Jerry Wei, Chengrun Yang, Xinying Song, Yifeng Lu, Nathan Hu, Jie Huang, Dustin Tran, Daiyi Peng, Ruibo Liu, Da Huang, Cosmo Du, and Quoc V. Le. 2024b. Long-form factuality in large language models. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA. Curran Associates Inc.

A Data Release

The SciELF dataset can be found at `https://anonymous.4open.science/r/SciELF-1A70/`.

B Model Usage

The Llama-3.3-70b-Instruct model is accessed through a resource hosting the model using vLLM. The LiteLLM library is used to call the model. The GPT-4.1 series of models were accessed via the OpenAI API. The generation parameters for both services are: *temperature* is set to 0 and *top_p* is 0.95.

C Results on the Synthetic Portion

Table 5 reports the atom-level performance on the synthetic portion of SciELF (868 instances). The ground-truth atomic labels for this split are derived automatically using GPT-4.1, which was provided with full PDF access and the exact details of the applied corruptions. For completeness, we report results across the original, corrupted, and combined splits.

The overall system rankings and values closely mirror those observed on the manually-curated portion. In the in-domain setting, FactReasoner leads across most metrics, achieving the highest combined F1 (0.861) and recall (0.867). Meanwhile, VERIFY attains the highest precision (0.912 on the combined split), which is consistent with its conservative labeling tendency.

This trend continues in the Web retrieval condition, where FactReasoner again achieves the highest recall and F1, while VERIFY maintains the highest precision on the corrupted and combined splits. Notably, the performance drop from in-domain to Web retrieval is pronounced across all systems. For instance, FactReasoner's combined F1 falls from 0.861 to 0.731, highlighting the increased difficulty associated with verifying scientific claims using open retrieval.

D Analysis of SciELF Continued

D.1 Manual Analysis on Corruption Type

We examine how FactReasoner responds to two corruption types: numeric alterations and context corruption, using concrete examples drawn from the manually annotated set.

Numeric alteration We find that FactReasoner can usually detect simple numeric corruptions. For example, when the aboveground biomass saturation point of WorldView-3 is corrupted from 247 Mg ha^{-1} to 189 Mg ha^{-1}, the retriever recovers the source passage that mentions the original value, and the verifier correctly predicts the atom "*WorldView-3 has a saturation point at 189 Mg ha*$^{-1}$" as NS. A more complex case reveals the verifier's difficulty in separating a specific numeric claim from its general meaning. The atom "*Hyperparameter tuning improved the Random Forest model's* R^2 *from* 0.19 *to* 0.60" is corrupted (where the true upper bound was 0.41). Evidence retrieved from another atom states that, "*The researchers compared the performance of different models, . . . , and found that grid tuning improved the* R^2 *and reduced the RMSE for all three groups.*" The verifier correctly infers that tuning helped, but because there was no context retrieved to negate the altered 0.60 value, the atom was incorrectly classified as supported.

Context corruption Context corruption involves distorting factual information found in the source paper (Section 3.1). In one instance, a corruption introduces the following false span: "*Image-derived vegetation indices were the dominant predictors in the optimal biomass model, accounting for over 70% of selected variables, while LiDAR height metrics contributed minimally to model performance.*" All atoms derived from this corrupted passage were correctly predicted as not supported (NS). The verifier successfully deduced this by retrieving the true context: "*Of the most important variables from Group-1 RF models, 68% out of the 28 selected variables were LiDAR-derived while 18% and 14% of the selected variables were derived from NAIP and Sentinel-2, respectively.*" This correct prediction aligns with the domain knowledge that forest aboveground biomass correlates strongly with canopy height and structural complexity, meaning LiDAR-based predictors naturally outperform image-based ones.

A corruption in a study on the simulation of hurricane rainfall swaps a directional attribute between two closely compared models. In the source article, the BEP model is explicitly noted to displace rainfall *westward*, while UCM accurately captures observed rainfall peaks. The corruption inverts this, yielding the false atom: "*UCM tended to displace rainfall westward for Hurricane Florence*." The retriever recovers a passage "*The UCM model produces higher 10 m local winds than the BEP model, which is associated with drag effects, and this leads to changes in the location and intensity of rainfall bands*" The verifier supports the statement due to the changes in location, but fails to isolate the error that westward displacement is attributable to the BEP model.

E Dataset Example

An instance of an original and corrupted synthetic example is given in Table 6.

F Prompts

To generate the synthetic portion of our dataset, we instruct the LLM to generate the Question-Answer pairs using the global summary of the paper. The prompt template used in our pipeline is presented in Listing 1. To predict the labels for the synthetic portion of the dataset, we include all corruption details alongside the PDF and instruct the LLM to predict the label of each atom. The prompt template is given in Listing 2.

Table 5: Atom-level performance on the synthetic portion, macro-averaged over instances. Ground truth labels are predicted by GPT-4.1. Best result per split is in **bold**.

Retrieval	System	Split	Accuracy	Precision	Recall	F1
In-Domain	FactReasoner	Original	**0.900**	1.000	**0.900**	**0.942**
		Corrupted	**0.825**	0.780	**0.834**	**0.780**
		Combined	**0.862**	0.890	**0.867**	**0.861**
	FActScore	Original	0.744	1.000	0.744	0.843
		Corrupted	0.770	0.775	0.702	0.712
		Combined	0.757	0.887	0.723	0.778
	VERIFY	Original	0.738	1.000	0.738	0.838
		Corrupted	0.790	**0.823**	0.695	0.732
		Combined	0.764	**0.912**	0.717	0.785
Web	FactReasoner	Original	**0.713**	1.000	**0.713**	**0.807**
		Corrupted	**0.702**	0.710	**0.678**	**0.655**
		Combined	**0.708**	**0.851**	**0.696**	**0.731**
	FActScore	Original	0.555	1.000	0.555	0.688
		Corrupted	0.647	0.696	0.544	0.578
		Combined	0.601	0.843	0.550	0.633
	VERIFY	Original	0.518	1.000	0.518	0.653
		Corrupted	0.664	**0.721**	0.524	0.562
		Combined	0.595	**0.851**	0.521	0.605

Question: What physical mechanisms were identified as responsible for the X-ray and radio signatures observed during the 2011 event in IGR J12580+0134?	
Corruption Type	External knowledge injection
Original	The physical mechanisms responsible for the X-ray and radio signatures observed during the 2011 event in IGR J12580+0134 were linked to a **tidal disruption event (TDE)**, where a supermassive black hole partially disrupted a super-Jupiter or star. This process generated a heavily absorbed power-law X-ray spectrum with a photon index of approximately 2.35 and a significant increase in X-ray flux. The radio signatures were interpreted as evidence of jet activity interacting with the circumnuclear medium, further supporting the TDE scenario with relativistic outflows.
Corrupted	The physical mechanisms responsible for the X-ray and radio signatures observed during the 2011 event in IGR J12580+0134 were primarily attributed to a **gamma-ray burst (GRB)** event, which is known to produce similar high-energy emissions and jet activity in galactic nuclei. The observed X-ray spectrum and radio signatures were thus interpreted as typical outcomes of GRB-induced relativistic jets interacting with the surrounding medium.
Domain Knowledge: *The corruption substitutes the true source mechanism (TDE) with a superficially similar but physically distinct phenomenon (GRB), exploiting the overlapping phenomenology of high-energy astrophysical events.*	

Table 6: Example of synthetic corruption from the SCIELF dataset. The corrupted passage injects external knowledge (GRB) that contradicts the source paper's mechanism (TDE).

Listing 1: System prompt used to instruct GPT-4.1 for synthetic QA generation and factual corruption.

```
You are given a structured global summary of a scientific paper:

---
{summary_text}
---

Your task:

1. Generate one scientifically valid question about the content of this paper.
2. Provide a *factual answer* fully supported by the paper.
3. Provide a *long-form comprehensive explanation* of the answer, grounded in the paper.
4. Now **corrupt the long-form answer** using the corruption type: **{corruption_type.value}**.
   - You MUST apply this corruption type in a scientifically meaningful way.
   - The corruption should be subtle but detectable by a domain expert.
5. Provide an explanation ("domain_knowledge") describing *how* the corruption is manifested,
   i.e., what domain knowledge a subject-matter expert would use to recognize the problem.
6. Identify whether the long-form answer contains citations (e.g., references, author-year, numeric footnotes).

Return only the JSON fields defined by this schema:
- question
- factual_answer
- long_form_comprehensive_answer
- falsified_or_corrupted_passage
- type_of_corruption
- domain_knowledge
- text_contains_citations
```

Listing 2: Prompt used to instruct GPT-4.1-mini for atom-level factual labelling.

```
You are evaluating atomic claims extracted from a scientific passage for factual correctness.

QUESTION:
{question}

GROUND-TRUTH FACTUAL ANSWER:
{factual_answer}

CORRUPTED PASSAGE (contains deliberately introduced errors):
{corrupted_answer}

CORRUPTION TYPE: {type_of_corruption}

EXPERT DESCRIPTION OF THE CORRUPTION:
{domain_knowledge}

[Optional] SOURCE PAPER (full text):
{document_text}

TASK:
For each atom below, classify it as:
  "S"  -- Supported: the claim is factually correct and consistent with the source paper
  "NS" -- Not-Supported: the claim contains or depends on the introduced error

Use the expert description of the corruption as your primary guide -- if an atom reflects
the introduced error it is NS, otherwise S. Use the source paper to verify specific
values, names, and relationships.

ATOMS:
{atoms}

Respond with a JSON object mapping each atom id to its label, e.g.:
{"a0": "NS", "a1": "S", "a2": "S"}

Only output the JSON object, nothing else.
```