DIVER: Enhancing Complex Fact Verification via Dynamic Evidence Retrieval and Iterative Reasoning

Anonymous ACL submission

Abstract

Fact-verification tasks involving sequences of claims remain challenging due to high claim density, low accuracy in open-domain evidence retrieval, and multi-hop reasoning requirements, which are difficult to address using traditional methods. In this paper, we propose **DIVER (Dynamic and Iterative fact VERification)**, a fact verification framework that decomposes paragraphs into context-independent sentences and applies a dynamic and iterative claim extraction and evidence retrieval strategy. Unlike prior one-shot or list-style approaches, DIVER introduces a fine-grained iterative claim extraction mechanism, allowing the system to better capture verifiable atomic claims, and incorporates a novel evidence-filtering and query recommendation module to robustly handle insufficient or ambiguous evidence, significantly enhancing multi-hop reasoning capabilities. Additionally, we propose a heuristic-driven revision step to detect long-distance contextual errors overlooked by previous approaches. These mechanisms collectively improve the model's calibration, ensuring the verifier fires only when supported by sufficient evidence—an essential property for dependable fact checking. Experimental results on three widely-used challenging fact-checking benchmarks (FEVEROUS, LIAR, and AVeriTeC) demonstrate that DIVER substantially outperforms existing LLM-based approaches and pipelines.

1 Introduction

Automated fact verification is an essential component of natural language processing, playing a critical role in combating misinformation and ensuring information quality across various applications such as news media, social networks, and automated content moderation systems (Guo et al., 2022). Among different scenarios, verifying claims embedded in contextually rich, semantically dense paragraphs presents unique challenges due to high claim density, ambiguous contexts, and requirements for multi-hop reasoning over large-scale, open-domain evidence sources (Jiang et al., 2020).

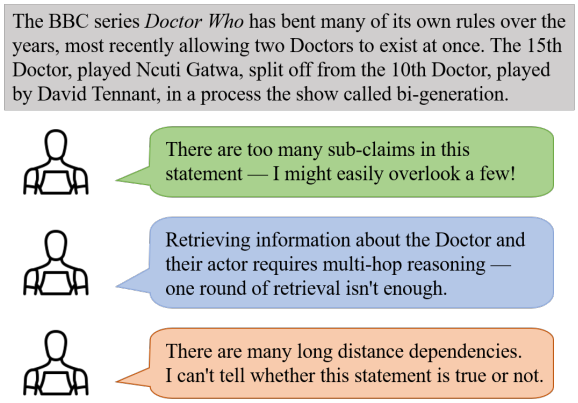

Figure 1: An real example in *um, actually* illustrating the three challenges current method faces: (1) claim density, (2) multi-hop evidence, and (3) long-distance context.

Benefiting from advanced semantic understanding and reasoning abilities of Large Language Models(LLMs), current mainstream fact-checking methods leverage LLMs to conduct decomposing, questioning and reasoning progressively, known as LLM-based methods (Vykopal et al., 2024). Despite their impressive contribution, these methods still face several limitations in practice, particularly with regard to claim extraction integrity, open-domain evidence retrieval accuracy and multi-hop reasoning (Deng et al., 2024; Zheng et al., 2024). First, existing methods typically adopt a static claim extraction strategy, where an LLM attempts to extract all verifiable claims from a sentence in a single pass. This approach often suffers from incompleteness, especially when dealing with complex sentences that contain multiple intertwined facts (Metropolitansky and Larson, 2025). As a result, some verifiable claims may be overlooked or under-specified (Metropolitansky and Larson, 2025). Second, open-domain evidence retrieval remains a significant bottleneck. Current methods often struggle with low retrieval accuracy, which limits the relevance and usefulness of the collected evidence (Zheng et al., 2024). Moreover, when multi-hop reasoning is required, these systems lack the ability to dynamically retrieve supporting evidence for subsequent reasoning steps (Zhuang et al.,

2024). For instance, even if the first-hop information is successfully retrieved, the system may fail to formulate effective follow-up queries to obtain second-hop evidence.

To overcome these limitations, we introduce DIVER (**D**ynamic and **I**terative fact **VER**ification), a verification framework specifically designed for context-rich, high-density paragraphs. DIVER first decomposes each paragraph into context-independent sentences and then performs dynamic, fine-grained claim extraction in an iterative loop: at each step, the model extracts exactly one verifiable atomic claim, generates a tailored query, and retrieves evidence from open-domain sources. A novel evidence-filtering + query-recommendation module evaluates the adequacy of retrieved documents and, when evidence is insufficient, autonomously formulates follow-up queries—enabling effective multi-hop reasoning. Finally, a heuristic-driven revision stage revisits the entire paragraph together with already-verified claims to surface long-distance contextual errors that single-sentence processing might miss.

The iterative loop in **DIVER** echoes the incremental question-answering paradigm of Quiz Bowl, where systems must decide *when* to answer as clues accrue rather than after reading the entire question. Prior work shows that making predictions only once sufficient evidence is accumulated improves both accuracy and model calibration (Boyd-Graber et al., 2012; He et al., 2016; Rodriguez et al., 2019). Analogously, DIVER refrains from issuing a SUPPORTED/REFUTED verdict until its evidence-retrieval loop has gathered enough support, yielding a more cautious—and ultimately more reliable—fact-checking system.

Our main contributions are:

- **Iterative claim extraction.** DIVER extracts one atomic claim at a time, yielding a complete and precise claim set.
- **Evidence-aware multi-hop retrieval.** A query-and-filter loop prunes noise and auto-generates follow-up queries, enabling accurate multi-hop verification.
- **Paragraph-level revision.** A final heuristic pass over the full paragraph surfaces long-distance errors missed at sentence level.

2 Related Work

Complex-claim fact verification. Early large-scale benchmarks such as FEVER (Thorne et al., 2018) and LIAR (Wang, 2017) sparked considerable interest in automatic fact checking, yet their claims are typically short and syntactically simple (Eisenschlos et al., 2021). Subsequent datasets have progressively raised the bar: FEVEROUS (Aly et al., 2021b) augments FEVER with tables, lists, and longer passages; AVeriTeC (Schlichtkrull et al., 2023) introduces paragraph-level claims that require multi-hop reasoning across open-domain sources; and SCIFACT (Wadden et al., 2020) focuses on scientific abstracts with domain shift. To cope with the resulting complexity, researchers have explored sentence decomposition (Liu et al., 2020), claim segmentation (Chen et al., 2022), and evidence graph construction (Chen et al., 2021). Nonetheless, most pipelines still treat decomposition as a one-shot preprocessing step, leaving them vulnerable to missing or underspecified atomic claims—a gap DIVER addresses with its fine-grained iterative extraction loop.

LLM-based verification methods. The rise of instruction-tuned large language models (LLMs) has turned fact-checking pipelines toward prompt-driven reasoning. Early work combines retrieval-augmented generation (RAG) (Lewis et al., 2020) with chain-of-thought prompting (Wei et al., 2022) or self-ask-with-search (Press et al., 2023) to guide LLMs through a coarse "ask–search–verify" loop. Subsequent systems such as UL2R (Tay et al., 2023) and RARR (Gao et al., 2023) integrate external retrievers but still rely on a static set of claims and fixed query plans, leaving them vulnerable when initial evidence is noisy or when multi-hop queries are required. More recently, a line of work relies on carefully designed prompt chains to decompose problems and guide evidence search: HISS (Zhang and Gao, 2023) introduces a hierarchical, step-wise set of prompts that first reformulate news claims and then iteratively retrieve and validate evidence; FactCheck-GPT (Wang et al., 2024) employs a multi-stage "ask-search-verify" prompting template to evaluate generation fidelity; and BiDeV (Liu et al., 2024) alternates "vagueness-defusing" and "redundancy-defusing" prompts to rewrite claims and filter noisy evidence before final judgment. Although these approaches may trigger multiple retrieval queries, one for each fragment of the claim—they still perform claim extraction in a single shot; and each query is handled in a single round, without being revised in light of newly found evidence, a design that often yields shallow or broken multi-hop evidence chains.

3 Method

We present **DIVER** (Dynamic and Iterative fact VERification), a modular framework *inspired by professional fact-checking workflows* while remaining fully automatable with LLMs and standard IR components.

Figure 2 gives a schematic view, and Algorithm 1 lists the high-level procedure.

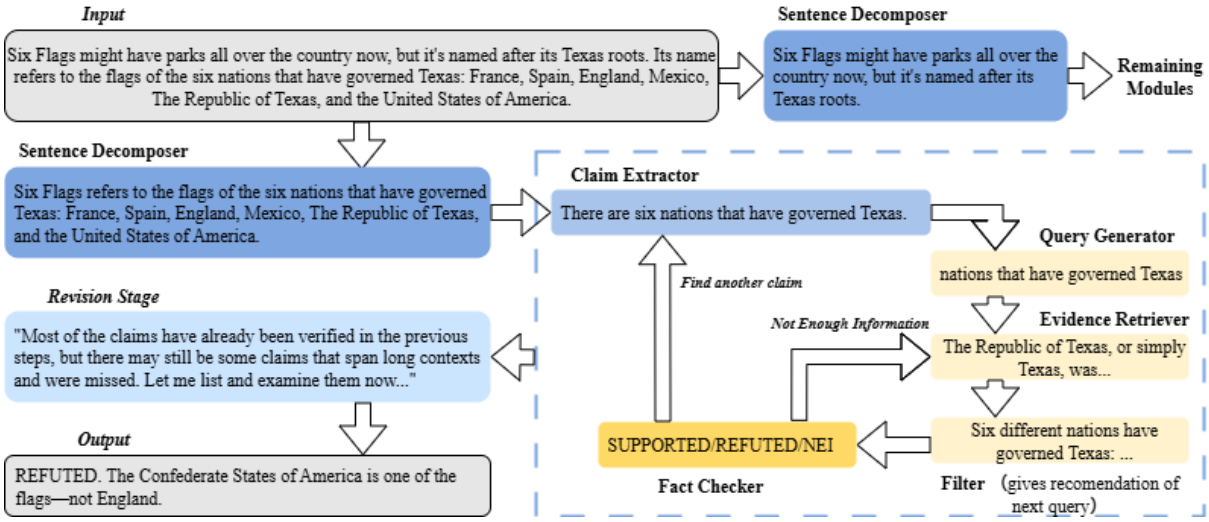

Figure 2: Illustration of the **DIVER** workflow. Given an input paragraph, the *Sentence Decomposer* rewrites each sentence so that it no longer relies on surrounding context (blue boxes, top). Each reformulated sentence enters the dashed *iterative loop*: a *Claim Extractor* selects *one* atomic claim, a *Query Generator* creates a search query, the *Evidence Retriever* returns passages, and a *Filter* removes noise while proposing a follow-up query. The *Fact Checker* labels the claim as SUPPORTED, REFUTED, or NEI. If the label is NEI, the recommended query is issued and the loop repeats; otherwise the next claim is extracted. After all sentences have been processed, a *Revision Stage* (left column) revisits the full paragraph plus previously-verified claims to discover long-distance errors that may have been missed. Finally, DIVER aggregates all verdicts to produce a paragraph-level decision.

3.1 Sentence Decomposer

The *Sentence Decomposer* converts a discourse-level paragraph $P = \{s_1, \ldots, s_m\}$ into a set of *context-independent* sentences $\mathcal{U} = \{u_1, \ldots, u_m\}$ through three guided edits:

1. **Coreference grounding.** The model resolves personal pronouns (*he, she*), demonstratives (*this, those*), and zero anaphora occurring at sentence boundaries, replacing them with their nearest explicit referents.
2. **Implicit-argument recovery.** Temporal or locative modifiers that are implicit from previous context (*"last month", "the capital"*) are made explicit (*"in March 2024", "the capital of France"*). This removes hidden dependencies that would otherwise leak information between sentences.
3. **Minimal-edit rewriting.** Finally, the sentence is re-serialised while keeping its token order and wording as intact as possible, thereby preserving the original claim surface form for later alignment with retrieved evidence.

The following example illustrates the three edits step-by-step.

Raw sentence: "He became president in 1999 and moved to the capital the following year." t**1. Coreference grounding:**
"Nelson Mandela became president in 1999 and moved to the capital the following year." **2. Implicit-argument recovery:**
"Nelson Mandela became president of South Africa in 1999 and moved to the capital, Pretoria, the following year."
3. Minimal-edit rewrite:
"Nelson Mandela became president of South Africa in 1999 and moved to Pretoria in 2000."

We perform all three edits with a single LLM call per sentence, adding only 8–10 ms latency. In our pilot study on 200 randomly sampled paragraphs from FEVEROUS, the normalised sentences boost downstream claim-extraction recall by 4–6 pp compared with leaving the original discourse unchanged.

3.2 Iterative and Incremental Claim Verification Loop

For every $u \in \mathcal{U}$ we launch an iterative loop (lines 2–14 in Algorithm 1).

Claim Extractor (M_c). The LLM executes a step–by–step reasoning routine to surface the next verifiable unit of knowledge under two guiding principles: (1) a proposition is deemed *atomic* when it cannot be further decomposed without altering its truth-conditional content; (2) a proposition is considered *novel* when it is not subsumed by the set of claims already extracted in earlier iterations. Formally, the i-th claim is given by

$$c_i = M_c\big(u, \{c_1, \ldots, c_{i-1}\}\big), \qquad (1)$$

where M_c is the claim-extraction module. Equation (1) enumerates the sentence's full slate of factual commitments while preventing any conflation of distinct propositions.

Query Generator (M_q). Starting from the freshly distilled claim c_i, the LLM crafts a retrieval query by (i) isolating salient lexical cues—named entities, temporal anchors, relational predicates—and (ii) composing them into a canonical search string enriched with discriminative domain keywords. The resulting query is

$$q_i = M_q(c_i), \tag{2}$$

where M_q maps a claim to its search query. Equation (2) aims to maximise the chance of retrieving evidence that is both topically relevant and diagnostic for the claim's truth value.

Search Module ($\mathcal{R}$). The query q_i is sent to an open-domain retriever $\mathcal{R}$, which returns a ranked list of passages,

$$\mathcal{D}_i = \mathcal{R}(q_i), \tag{3}$$

forming a candidate evidence pool that is both topically aligned with and potentially diagnostic for the claim.

Filter & Query Recommendation (M_f). Given the candidate set $\mathcal{D}_i$ and the claim c_i, the evidence-filter module M_f executes two complementary actions: (i) it prunes off-topic or redundant passages to distil a concise evidence bundle $\mathcal{E}_i$; and (ii) it composes a follow-up query q_i^{rec} for any still-missing links, such that

$$(\mathcal{E}_i,\, q_i^{\text{rec}}) = M_f\big(\mathcal{D}_i, c_i\big). \tag{4}$$

Equation (4) therefore encapsulates both evidence selection and query recommendation in a single step.

The filtered evidence $\mathcal{E}_i$ is passed to the *Fact Checker*, whereas q_i^{rec} is only executed if the checker later returns NOT ENOUGH INFORMATION, enabling graceful multi-hop escalation without superfluous retrieval calls.

Fact Checker (M_{ch}). The pair $\langle c_i, \mathcal{E}_i \rangle$ is fed into a fact-checking LLM prompt, which returns a verdict $v_i \in \{\text{SUPPORTED}, \text{REFUTED}, \text{NOT ENOUGH INFORMATION}\}$ together with a concise rationale ρ_i, such that

$$(v_i,\, \rho_i) = M_{ch}\big(c_i, \mathcal{E}_i\big). \tag{5}$$

Equation (5) formalises the final entailment step that maps each claim-evidence pair to a label and justification.

Should v_i be NOT ENOUGH INFORMATION, the system immediately re-enters retrieval with the follow-up query q_i^{rec}; otherwise the triple $\langle c_i, v_i, \rho_i \rangle$ is persisted as a verified claim.

Algorithm 1 DIVER Framework

Require: Paragraph P
Ensure: Label $y \in \{\text{SUPPORTED}, \text{REFUTED}\}$, explanation ξ

1: $\mathcal{U} \leftarrow \text{SENTENCEDECOMPOSER}(P)$
2: **for each** $u \in \mathcal{U}$ **do**
3: **while true do**
4: $c \leftarrow \text{CLAIMEXTRACTOR}(u)$
5: $q \leftarrow \text{QUERYGENERATOR}(c)$
6: $\mathcal{D} \leftarrow \text{RETRIEVE}(q)$
7: $\mathcal{E},\, q^{\text{rec}} \leftarrow \text{FILTER}(\mathcal{D}, c)$
8: $v, \rho \leftarrow \text{FACTCHECKER}(c, \mathcal{E})$
9: **if** $v = \text{NEI}$ **then**
10: $q \leftarrow q^{\text{rec}}$; **continue**
11: **else**
12: store (c, v, ρ); **break**
13: **end if**
14: **end while**
15: **end for**
16: **for** $t = 1$ **to** $r_{\max}$ **do**
17: $c^{\text{rev}} \leftarrow \text{REVISIONEXTRACTOR}(P, \mathcal{C})$
18: **if** $c^{\text{rev}} = \varnothing$ **then break**
19: **end if**
20: verify c^{rev} via lines 2–14
21: **end for**
22: $y, \xi \leftarrow \text{AGGREGATE}(\mathcal{C})$
23: **return** y, ξ

3.3 Revision Stage

After all sentences finish their loops, a *Revision* stage revisits the entire paragraph. The LLM sees P and the set of already-verified claims $\mathcal{C}$, then proposes an additional claim c^{rev} if it suspects a long-range inconsistency. The same extract–search–filter–check cycle operates during the revision stage. At iteration t, the revision module M_r proposes an additional claim

$$c_t^{\text{rev}} = M_r\big(P, \mathcal{C}_{t-1}\big), \qquad t = 1, \ldots, r_{\max}, \tag{6}$$

where the default cap is $r_{\max} = 3$. The loop terminates once the marginal gain in uncovered errors falls below a preset threshold or when $t = r_{\max}$.

3.4 Decision Aggregation

A final aggregation module M_a maps the set of verified claims $\mathcal{C}$ to a paragraph-level verdict y and a composite explanation ξ:

$$(y,\, \xi) = M_a\big(\mathcal{C}\big). \tag{7}$$

The paragraph is labelled SUPPORTED iff *all* claims in $\mathcal{C}$ are SUPPORTED; otherwise it is REFUTED. When refuted, ξ concatenates the rationales returned by the fact-checker for every non-supported claim.

4 Experiments

Evaluation overview. Our empirical study addresses three questions: *(Q1)* How does **DIVER** compare with state-of-the-art LLM-based pipelines on standard fact-checking benchmarks of differing density and reasoning depth? *(Q2)* Which components—iterative extraction, evidence-aware multi-hop retrieval, and paragraph-level revision—contribute most to its effectiveness? *(Q3)* Can the framework generalise to real-world, noisy inputs beyond curated datasets?

To this end we evaluate DIVER and a suite of strong baselines on **three public benchmarks** (FEVEROUS, LIAR, AVeriTeC), plus a *case-study* set from the live quiz show *Um, Actually*. All systems share the same open-web retrieval backend (Google Custom Search) and are run on two backbone models—*GPT-4o-mini* and the weaker *GPT-3.5-turbo*—so that *improvements stem solely from pipeline design*. Following prior work, we report paragraph-level **Accuracy**; additional breakdowns (Macro-F1, error taxonomy, cost) appear in later sections.

Section 4 first details datasets, baselines, and implementation choices. We then present overall results (Table 1), component ablations, policy studies, multi-hop evaluation on HoVer, and a qualitative error analysis that pinpoints the remaining failure modes.

4.1 Experimental Setup

Datasets. We evaluate DIVER on three widely–used, challenging fact-checking benchmarks to evaluate the fact-checking performance of the baselines and DIVER: *(i) FEVEROUS* (Aly et al., 2021a), *(ii) LIAR* (Wang, 2017), and *(iii) AVeriTeC* (Schlichtkrull et al., 2023).

Baselines. To demonstrate the effectiveness of our approach, we compare against four categories of baselines: *(i) Zero-retrieval LLM*: Vanilla GPT model without external evidence; *(ii) LLM+web search*: GPT model with Google web search; *(iii) Retrieval-plugged pipelines*: UL2R (Tay et al., 2022) and RARR (Gao et al., 2023), which insert external retrievers into fixed prompt templates; *(iv) Step-by-step prompting methods* (our direct competitors): HISS (Zhang and Gao, 2023), FactCheck-GPT (Wang et al., 2024), and BiDeV (Liu et al., 2024). All baselines share the same GPT model backend and web retrieval to ensure fairness.

Evaluation Metrics. We report paragraph-level **Accuracy** for the labels SUPPORTED, REFUTED, and NEI.

Implementation Details. All methods, including our own and every baseline, rely on *the same* open-web retrieval backend: the Serper.dev API,[1] which returns live Google Search result pages (SERP) in JSON format. For each query we request the top–10 hits, cache the returned URLs, and download the corresponding pages with `requests` plus Readability parsing; only the main textual content is retained to comply with the API's Terms of Service. Every LLM call—*Extractor*, *Query Generator*, *Filter*, *Fact Checker*, and baseline pipelines—uses the identical `gpt-4o` model (temperature 0.3, top-p 0.95). Prompts and other hyperparameters are tuned on the development split of each dataset; the complete prompt list and caching script are provided in Appendix A and our public code repository.

4.2 Overall Performance

The evaluation of DIVER and the baselines is conducted on three benchmarks, based on two backbone models: GPT-4o-mini and GPT-3.5-turbo. The experimental results are presented in Table 1, from which we can draw the following analysis.

4.3 Main Results Analysis

4.3.1 Main Results Analysis

As shown in Table 1, **DIVER** consistently achieves the best performance across all three benchmarks and both backbone LLMs. Compared with the strongest step-wise baseline, BiDeV, DIVER gains **+5.7** / **+7.2** points on FEVEROUS, **+0.0** / **+2.7** on LIAR, and **+3.6** / **+10.8** on AVeriTeC when using GPT-4o-mini and GPT-3.5-turbo respectively. These improvements correlate with task complexity: FEVEROUS and AVeriTeC involve densely packed claims and multi-hop reasoning, offering more opportunities for DIVER's dynamic claim decomposition and revision to take effect. In contrast, LIAR mainly contains short, single-hop claims, where DIVER's iterative mechanism is underutilized.

DIVER's relative advantage is even more pronounced on the weaker backbone (*GPT-3.5-turbo*), averaging **+6.0** points over BiDeV compared to **+3.3** on GPT-4o-mini. This suggests that DIVER's structured control—especially its adaptive retrieval and revision loop—can effectively compensate for the limited reasoning and instruction-following ability of smaller models. On stronger models like GPT-4o-mini, DIVER still yields substantial gains, indicating that even advanced LLMs benefit from guided decomposition and targeted evidence alignment.

A breakdown across baselines reveals a clear performance hierarchy: HISS < FactCheck-GPT < BiDeV < DIVER. This reflects the cumulative benefits of modular design: each added component—fine-grained claim extraction, evidence

[1] `https://serper.dev`, accessed July 2025

filtering, query recommendation, and paragraph-level revision—yields incremental yet complementary improvements. While BiDeV integrates basic query planning, DIVER distinguishes itself by enabling query adaptation conditioned on retrieved evidence and revisiting claims at the paragraph level.

To further understand these performance trends, we conduct a detailed error analysis in Section 4.6. The results show that DIVER significantly reduces sub-claim omissions and retriever failures, especially on complex datasets. However, challenges remain in fine-grained entailment recognition, which we analyze in detail later.

4.4 Iteration–Policy Ablation

Both the claim–extraction loop (i iterations per sentence) and the revision loop (j iterations per paragraph) can be governed by a variety of stopping policies. We experiment with four alternatives for i and three for j:

- **CE-1** *Self-termination.* The extractor outputs a special STOP token once it believes no novel atomic claim remains.once it believes no novel atomic claim remains.
- **CE-2** *Fixed budget n.* Grid-search on dev shows n=5 gives best performance.
- **CE-3** *Length-based.* $i = \lceil |u|/n \rceil$ with n=4 tokens; longer sentences receive proportionally more extraction steps.
- **CE-4** *Entity-based.* $i = n \times$ (number of named entities) with n=3, leveraging the intuition that each entity typically anchors at least one fact.
- **REV-1** *Fixed budget n* (n=4).
- **REV-2** *Slack budget.* $j = \max\big(n - |\mathcal{C}|, 0\big)$ with n=15.
- **REV-3** *Length–entity hybrid.* $j = \max\big(\lceil |P|/n \rceil - i\,|\mathcal{C}|, 0\big)$ where n=4 and per-claim discount i=1.

Discussion. **(i)** Allowing the LLM to self-terminate (CE-1) yields the best overall accuracy, verifying that the model can reliably decide when its coverage is complete. Both the constant-budget rule (CE-2) and the entity-triggered rule (CE-4) trail by roughly one point, while length-based CE-3 under-extracts on terse sentences and over-extracts on long, list-like sentences (numbers omitted for space). **(ii)** For revision, a simple fixed budget of four passes (REV-1) performs on par with the slack heuristic REV-2 but avoids maintaining a global claim counter and is therefore retained as our default. The hybrid rule REV-3, which ties the budget to paragraph length, proves too aggressive and often flips otherwise correct paragraphs into REFUTED. Overall, the CE-1 + REV-1 pair (highlighted in Table 2) offers the best balance between recall and precision without incurring excessive cost.

4.5 Ablation Study

To quantify the impact of each component in DIVER, we conduct a leave-one-out ablation study on the same three benchmarks. Table 3 reports paragraph-level *Accuracy*) when individual modules are removed or simplified while all other settings remain unchanged.

Iterative extraction vs. one-shot. Removing the step-wise claim extractor (*–Iterative Extraction*) causes a drop of **6.3** % on FEVEROUS and **7.9** % on AVeriTeC, confirming that fine-grained, incremental extraction is critical for high-density paragraphs.

Evidence filtering and query recommendation. Without the filter/recommend module (*–Filter / Recommend*), performance degrades most severely on multihop datasets (−10.4 on FEVEROUS, −7.2 on AVeriTeC), showing that adaptive follow-up queries are indispensable when initial retrieval is noisy or incomplete.

Revision stage. Skipping the paragraph-level revision (*–Revision*) impact on FEVEROUS(−3.3) and AVeriTeC(−4.2), indicating that revision is mainly useful for long-distance contextual errors.

Overall, each module contributes complementary gains, and their combination is required to achieve the best results reported in Section 4.

4.5.1 Effectiveness of Iterative Claim Extraction

To isolate the benefit of our *iterative* claim–by–claim extraction paradigm, we sampled 150 paragraph–length items from the REFUTED split of *AVeriTeC*. All samples contain a high density of sub-claims (avg. 6.4 per paragraph). For each paragraph we applied either (i) a standard *one-shot* extractor that outputs all claims in a single pass, or (ii) our *iterative* extractor that selects exactly one atomic claim per step until no new information can be found. Two annotators then judged whether the extracted set of claims *contains at least one erroneous sub-claim*—the necessary pre-condition for a downstream fact checker to recover the paragraph's REFUTED label.

Table 4 shows that iterative extraction raises coverage by **+7.3 pp** on GPT-4o-mini and **+9.2 pp** on GPT-3.5-turbo. The gain is more pronounced for the weaker backbone, confirming that step-wise focusing helps lower-capacity models surface subtle erroneous facts that a one-shot pass often merges or omits.

Methods	FEVEROUS		LIAR		AVeriTeC	
	4o-mini	3.5-turbo	4o-mini	3.5-turbo	4o-mini	3.5-turbo
Vanilla LLM	50.1	29.8	59.6	29.1	70.0	41.5
LLM + Web Search	55.7	42.7	65.2	40.2	74.1	60.6
HISS (Zhang and Gao, 2023)	59.3	48.2	58.6	46.8	68.2	44.9
FactCheck-GPT (Wang et al., 2024)	65.3	56.5	65.2	52.9	75.8	63.0
BiDeV (Liu et al., 2024)	65.9	59.5	67.4	60.3	79.4	64.4
DIVER (ours)	**71.6**	**66.7**	**67.4**	**63.0**	**83.0**	**75.2**

Table 1: Performance (% Accuracy *or* Macro-F1) of baseline methods and our DIVER on three fact-checking benchmarks. Each column pair shares the same retrieval backend (Google Search) and differs only in the underlying LLM. All metrics are computed from a single run.

Claim Extractor	Revision Stage	AVeriTeC	FEVEROUS
CE-1	REV-1	**83.0**	**71.6**
CE-1	REV-2	82.3	71.2
CE-2	REV-1	80.5	67.8
CE-2	REV-2	79.5	70.8
CE-4	REV-1	79.1	69.9
CE-4	REV-2	78.6	70.2
... other combinations omitted for brevity ...			

Table 2: Accuracy on AVeriTeC and FEVEROUS dev for representative iteration-policy pairs. CE-1 + REV-1 is selected as the default configuration.

Variant	FEV.	LIAR	AVT.
Full DIVER	**71.6**	**67.4**	**83.0**
– Iter. Extr.	65.3	66.8	75.1
– Filter / Rec.	61.2	63.1	75.8
– Revision	68.3	67.2	78.8

Table 3: Ablation results (%). Each row removes one component from the full system.

Extractor	GPT-4o-mini	GPT-3.5-turbo
One-shot	85.8%	76.3%
Iterative (ours)	**93.1%**	**85.5%**

Table 4: Percentage of paragraphs whose extracted claim set includes *at least one* erroneous sub-claim. Higher is better for triggering a correct REFUTED verdict.

Method	HoVer (hop-3)	HoVer (hop-4)
FactCheck-GPT (Wang et al., 2024)	60.11	59.25
FLAN-T5 (Jiang et al., 2021)	60.23	55.42
BiDeV (Liu et al., 2024)	63.62	60.41
DIVER (ours)	**65.48**	**62.82**

Table 5: Accuracy (%) on HoVer hop-3 / hop-4 (open-domain). All systems use the same Google-Search retriever and GPT-4o-mini backbone; DIVER alone employs dynamic follow-up queries.

4.5.2 Impact of Dynamic Multi-hop Retrieval

We next evaluate DIVER on the *open-domain* splits of **HoVer** (Jiang et al., 2020) for two challenging settings that *require* chained evidence: *hop-3* and *hop-4*.[2] Table 5 contrasts our results with representative step-by-step baselines.

Findings DIVER surpasses the strongest baseline (BiDeV) by **+1.9 pp** on hop-3 and **+2.4 pp** on hop-4. The margin widens as the evidence chain length grows, suggesting that our *Filter & Recommendation* loop is particularly effective when the initial retrieval misses intermediate links. Compared with single-hop-prompting systems such as FactCheck-GPT and FLAN-T5, DIVER yields gains of **+5.3–7.4 pp**, confirming that *adaptive multi-hop querying* is crucial for deep reasoning tasks.

4.6 Error Breakdown

To understand *how* DIVER improves over earlier pipelines, we randomly sampled **100** misclassified paragraphs from the FEVEROUS test set for both DIVER and the strongest baseline BiDeV, then manually assigned each error to one of five mutually exclusive categories (Table 6).

[2] A claim is labelled hop-k if at least k distinct evidence sentences must be concatenated to establish its truth.

Error Type	DIVER	BiDeV	Δ
Missed sub-claim (MSC)	12	27	−15
Retriever failure (RF)	18	34	−16
Uncaught refutation (UC)	14	17	−3
Spurious refutation (SR)	46	12	+34
Other	10	10	0
Total	100	100	

Table 6: Manual taxonomy of 100 erroneous predictions per system. MSC = the true contradictory sub-claim never extracted; RF = relevant evidence not retrieved; UC = evidence contradicts claim but checker outputs SUPPORTED/NEI; SR = evidence *does not* contradict claim but checker outputs REFUTED.

Findings. DIVER eliminates more than half of the *Missed sub-claim* errors and reduces *Retriever failures* by 47 %, confirming that iterative extraction and follow-up querying successfully plug the two largest gaps of one-shot pipelines. It is also slightly better at recognising genuine contradictions (UC, −3 errors).

The price we pay is an increased rate of *Spurious refutation* (SR): the fact checker sometimes over-trusts a narrow slice of evidence and flags an otherwise correct claim as REFUTED. We conjecture that the stronger recall of our loop delivers *more* borderline passages to the entailment model, amplifying its susceptibility to false negatives. Mitigating this tendency—e.g. via confidence calibration or ensemble voting—is left for future work.

4.7 Efficiency & Cost

On the FEVEROUS test set, our pipeline executes $\mathbf{10.2 \pm 1.8}$ serial[3] GPT-4o calls and $\mathbf{8.6 \pm 1.1}$ search-engine queries per paragraph.

These calls consume on average $\mathbf{8\,842 \pm 952}$ prompt tokens and $\mathbf{595 \pm 101}$ completion tokens.

For a full per-stage cost table and the token-counter script, see Appendix §A.5.

4.8 Real–world Case Study: *Um, Actually*

Motivation. Standard fact-checking corpora are carefully curated but often synthetic. To assess DIVER under truly *in-the-wild* conditions, we collected a set of multi-sentence corrections from the 2024 season of the quiz show *Um, Actually*.[4] Each item contains a host statement (3–5 sentences, dense with trivia) and at least one contestant–supplied correction.

Dataset construction. We extracted automatic subtitles, normalised them, and asked two expert annotators to label every atomic claim as SUPPORTED, REFUTED, or NEI. The resulting **210** paragraphs (average **6.8** sub-claims each) are released—subtitles and labels only—at https://github.com/*your-repo*/umactually-facts for reproducibility, thereby avoiding redistribution of copyrighted footage.

Results. Table 7 compares DIVER with two strong baselines that share the same Google-Search retriever and GPT-4o-mini backbone.

Method	Macro-F1	Acc.
Vanilla GPT-4o	32.3	31.9
BiDeV	74.2	71.8
DIVER (ours)	**80.6**	**79.5**

Table 7: Performance on the *Um, Actually* case-study set.

DIVER outperforms BiDeV by **+6.4** Macro-F1, mirroring the gains observed on HoVer hop-4. Qualitative inspection shows that our *Revision* stage frequently recovers errors spread across non-adjacent sentences, a pattern typical of live-spoken trivia.

Limitations. Because the dataset is derived from entertainment media and lacks third-party guidelines, we treat these results as *indicative* rather than conclusive; nevertheless, they highlight DIVER's robustness to noisy, real-world inputs.

5 Conclusion

We present **DIVER**, a fact-checking system that decomposes complex input into atomic claims and dynamically guides retrieval and verification through an adaptive revision loop. Unlike prior step-wise approaches, DIVER jointly optimizes claim extraction, query planning, and evidence aggregation, enabling more accurate and robust fact verification across challenging open-domain benchmarks.

Experiments on FEVEROUS, LIAR, and AVeriTeC demonstrate that DIVER consistently outperforms strong baselines, with the largest gains observed on tasks requiring dense, multi-hop reasoning. Analysis further reveals that DIVER's design particularly benefits weaker language models by supplying structured, high-quality evidence through targeted, iterative search.

Future work includes enhancing the robustness of entailment prediction, improving system efficiency, and extending DIVER to support real-time fact-checking in dynamic web environments.

References

Rami Aly, Zhijiang Guo, Michael Schlichtkrull, James Thorne, Andreas Vlachos, Christos Christodoulopoulos, Oana Cocarascu, and Arpit Mittal. 2021a. Feverous: Fact extraction and

[3] "Serial" = cannot be overlapped with retrieval or other LLM calls.

[4] https://www.youtube.com/c/umactually

verification over unstructured and structured information. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1.

Rami Aly, Zhijiang Guo, Michael Sejr Schlichtkrull, James Thorne, Andreas Vlachos, Christos Christodoulopoulos, Oana Cocarascu, and Arpit Mittal. 2021b. FEVEROUS: fact extraction and verification over unstructured and structured information. *CoRR*, abs/2106.05707. NeurIPS Datasets and Benchmarks Track.

Jordan Boyd-Graber, Brianna Satinoff, He He, and Hal Daumé III. 2012. Besting the quiz master: Crowdsourcing incremental classification games. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1290–1301, Jeju Island, Korea. Association for Computational Linguistics.

Chonghao Chen, Fei Cai, Xuejun Hu, Wanyu Chen, and Honghui Chen. 2021. HHGN: A hierarchical reasoning-based heterogeneous graph neural network for fact verification. *Inf. Process. Manag.*, 58(5):102659.

Jifan Chen, Aniruddh Sriram, Eunsol Choi, and Greg Durrett. 2022. Generating literal and implied subquestions to fact-check complex claims. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3495–3516, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Zhenyun Deng, Michael Schlichtkrull, and Andreas Vlachos. 2024. Document-level claim extraction and decontextualisation for fact-checking. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11943–11954, Bangkok, Thailand. Association for Computational Linguistics.

Julian Eisenschlos, Bhuwan Dhingra, Jannis Bulian, Benjamin Börschinger, and Jordan Boyd-Graber. 2021. Fool me twice: Entailment from Wikipedia gamification. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 352–365. Association for Computational Linguistics.

Luyu Gao, Zhuyun Dai, Panupong Pasupat, Anthony Chen, Arun Tejasvi Chaganty, Yicheng Fan, Vincent Y. Zhao, Ni Lao, Hongrae Lee, Da-Cheng Juan, and Kelvin Guu. 2023. RARR: Researching and revising what language models say, using language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL 2023)*, pages 16477–16508, Toronto, Canada.

Zhijiang Guo, Michael Schlichtkrull, and Andreas Vlachos. 2022. A survey on automated fact-checking. *Transactions of the Association for Computational Linguistics*, 10:178–206.

He He, Jordan Boyd-Graber, Kevin Kwok, and Hal Daumé III. 2016. Opponent modeling in deep reinforcement learning. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1804–1813, New York, New York, USA. PMLR.

Kelvin Jiang, Ronak Pradeep, and Jimmy Lin. 2021. Exploring listwise evidence reasoning with t5 for fact verification. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 402–410, Online. Association for Computational Linguistics.

Yichen Jiang, Shikha Bordia, Zheng Zhong, Charles Dognin, Maneesh Singh, and Mohit Bansal. 2020. HoVer: A dataset for many-hop fact extraction and claim verification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3441–3460, Online. Association for Computational Linguistics.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Yuxuan Liu, Hongda Sun, Wenya Guo, Xinyan Xiao, Cunli Mao, Zhengtao Yu, and Rui Yan. 2024. Bidev: Bilateral defusing verification for complex claim fact-checking. *arXiv preprint arXiv:2502.16181*.

Zhenghao Liu, Chenyan Xiong, Maosong Sun, and Zhiyuan Liu. 2020. Fine-grained fact verification with kernel graph attention network. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7342–7351, Online. Association for Computational Linguistics.

Dasha Metropolitansky and Jonathan Larson. 2025. Towards effective extraction and evaluation of factual claims. In *ACL 2025 Main Conference*.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah Smith, and Mike Lewis. 2023. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5687–5711, Singapore. Association for Computational Linguistics.

Pedro Rodriguez, Shi Feng, Mohit Iyyer, He He, and Jordan Boyd-Graber. 2019. Quizbowl: The case for incremental question answering. *CoRR*, abs/1904.04792.

Michael Schlichtkrull, Zhijiang Guo, and Andreas Vlachos. 2023. AVeriTeC: A dataset for real-world claim verification with evidence from the web. In *Advances in Neural Information Processing Systems 36 (NeurIPS 2023), Track on Datasets and Benchmarks*, pages 65128–65167. Curran Associates, Inc.

Yi Tay, Jason Wei, Hyung Chung, Vinh Tran, David So, Siamak Shakeri, Xavier Garcia, Steven Zheng, Jinfeng Rao, Aakanksha Chowdhery, Denny Zhou, Donald Metzler, Slav Petrov, Neil Houlsby, Quoc Le, and Mostafa Dehghani. 2023. Transcending scaling laws with 0.1% extra compute. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1471–1486, Singapore. Association for Computational Linguistics.

Yi Tay, Jason Wei, Hyung Won Chung, Vinh Q. Tran, David R. So, Siamak Shakeri, Xavier Garcia, Huaixiu Steven Zheng, Jinfeng Rao, Aakanksha Chowdhery, Denny Zhou, Donald Metzler, Slav Petrov, Neil Houlsby, Quoc V. Le, and Mostafa Dehghani. 2022. Transcending scaling laws with 0.1% extra compute. *arXiv preprint arXiv:2210.11399*.

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.

Ivan Vykopal, Matúš Pikuliak, Simon Ostermann, and Marián Šimko. 2024. Generative large language models in automated fact-checking: A survey.

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.

William Yang Wang. 2017. "liar, liar pants on fire": A new benchmark dataset for fake news detection. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 422–426, Vancouver, Canada. Association for Computational Linguistics.

Yuxia Wang, Revanth Gangi Reddy, Zain Muhammad Mujahid, Arnav Arora, Aleksandr Rubashevskii, Jiahui Geng, Osama Mohammed Afzal, Liangming Pan, Nadav Borenstein, Aditya Pillai, Isabelle Augenstein, Iryna Gurevych, and Preslav Nakov. 2024. Factcheck-bench: Fine-grained evaluation benchmark for automatic fact-checkers. *arXiv preprint arXiv:2311.09000*. V3, April 2024.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.

Xuan Zhang and Wei Gao. 2023. Towards LLM-based fact verification on news claims with a hierarchical step-by-step prompting method. In *Proceedings of IJCNLP–AACL 2023*, pages 996–1011, Taipei, Taiwan.

Liwen Zheng, Chaozhuo Li, Xi Zhang, Yu-Ming Shang, Feiran Huang, and Haoran Jia. 2024. Evidence retrieval is almost all you need for fact verification. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 9274–9281, Bangkok, Thailand. Association for Computational Linguistics.

Ziyuan Zhuang, Zhiyang Zhang, Sitao Cheng, Fangkai Yang, Jia Liu, Shujian Huang, Qingwei Lin, Saravan Rajmohan, Dongmei Zhang, and Qi Zhang. 2024. EfficientRAG: Efficient retriever for multi-hop question answering. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 3392–3411, Miami, Florida, USA. Association for Computational Linguistics.