Um, Actually: Timely and Calibrated Fact-Checking with DIVER

Senyi Zhang
Computer Science
Sun Yat-sen University
Guangzhou, China
zhsenyi@gmail.com

Tasnim Kabir
Computer Science
University of Maryland
College Park, USA
tkabir1@umd.edu

Jordan Boyd-Graber*
Computing and Data Science
Nanyang Technological
University, Singapore
jordan.ying@ntu.edu.sg

Abstract

Fact verification pipelines are often unrealistic: they assume full-input access, operate on isolated atomic claims, and overlook the central question of *when* a system should act under partial evidence. We introduce DIVER (Deliberative and Iterative fact VERification), an LLM-based framework for calibrated incremental fact-checking. DIVER incrementally extracts verifiable atomic claims from progressively revealed input, retrieves and refines evidence through iterative web search, and triggers a correction as soon as a claim is reliably refuted; if no such trigger occurs, it performs a passage-level revision over the full input to recover missed errors. Across standard fact-checking benchmarks, DIVER outperforms strong LLM-based verification pipelines, with especially large gains on retrieval-intensive settings. We further introduce UMACTUALLY, a real-world benchmark built from the Internet game *Um, Actually*, to evaluate not only whether a system finds an error, but whether it does so at the right time. On UMACTUALLY, DIVER reports the strongest end-to-end results while typically buzzing close to the earliest valid decision point, supporting its effectiveness for calibrated incremental fact-checking.[1]

1 Introduction

Real-world fact-checking often takes place under incremental exposure, such as in social media reply chains, spoken question answering, or live commentary. In such settings, there is often no true "full input": new information continues to arrive, and a verifier must decide not only whether some part of the input is false, but also when enough evidence has become available to justify acting.

At the same time, factual errors in these settings are often embedded in dense context rather than presented as isolated atomic claims. A post might say, "You know he was involved in the first war—he even gave a speech at the UN." Verifying this statement requires resolving the pronoun "he" (e.g., Colin Powell), inferring that the post refers to the 2003 invasion of Iraq, and checking whether the described events are accurate. The difficulty is that the claim is not fully interpretable in isolation: its referents and implied background must be reconstructed from context. Moreover, the factual error itself may be subtle, such as incorrectly characterizing the 2003 invasion of Iraq as the "first war" between the United States and Iraq.

Thus, realistic verifiers must (i) isolate refutation-critical *atomic* claims from dense text, (ii) retrieve and refine evidence across hops, and (iii) decide *when* the available evidence is sufficient to commit. This makes *calibration* central: a dependable system should defer judgment when evidence is still insufficient, rather than hallucinate or commit prematurely.

Recent LLM-based verification pipelines combine decomposition, retrieval, and reasoning, but they only partially address the requirements above. First, *one-shot* claim extraction is brittle for dense, context-dependent discourse, often yielding incomplete or under-specified atomic claims (Metropolitansky and Larson, 2025). Second, open-domain retrieval is noisy, yet most systems lack an explicit mechanism to assess *evidence sufficiency* and iteratively refine queries for multi-hop verification (Zheng et al., 2024; Zhuang et al., 2024). Third, most pipelines assume that the full input is available upfront, and thus do not support quizbowl-like incremental verification, where the input is revealed token by token and the system must decide under partial exposure whether to wait for more context or commit to an answer.

We introduce DIVER (**D**eliberative and **I**terative fact **VER**ification), an LLM-based pipeline for Quizbowl-like incremental fact-checking. In this

*† Work completed while at University of Maryland

[1] Code and datasets are available at https://github.com/Senyi-Zhang/DIVER.

Findings of the Association for Computational Linguistics: EMNLP 2026, pages 18092–18106
October 24-29, 2026 ©2026 Association for Computational Linguistics

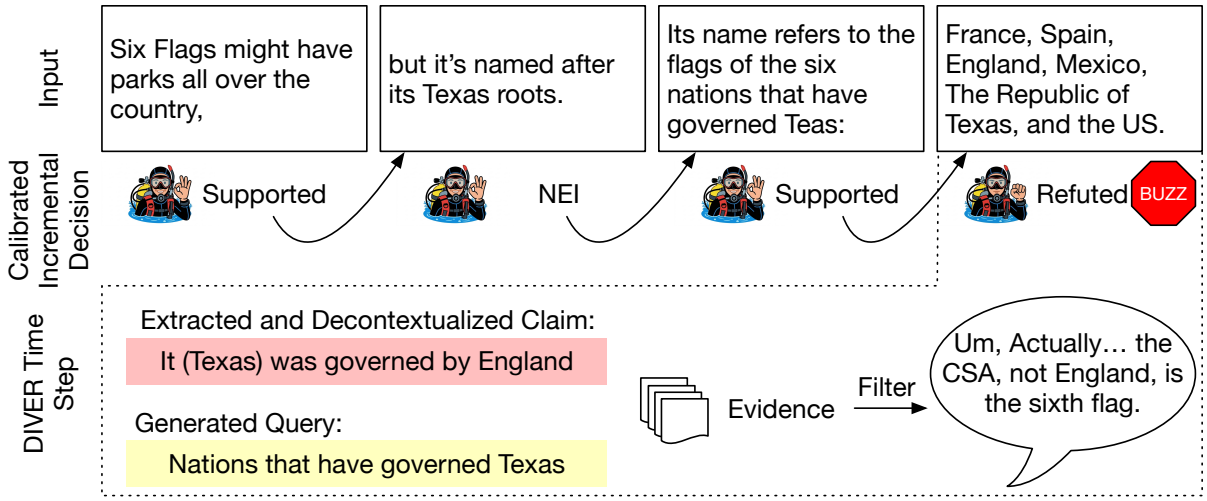

Figure 1: **Overview of DIVER.** The input is revealed token by token. As the passage unfolds, CLAIMEXTRACTOR proposes atomic claims, which enter an iterative retrieval-and-checking loop. DIVER buzzes as soon as a claim is verified as REFUTED; otherwise, if no such trigger occurs by the end of the passage, it performs passage-level revision.

setting, the input is revealed token by token, and the system may answer at any time. As a result, the core challenge is not only to determine whether the content is correct, but also to decide whether the currently available information is sufficient to justify commitment under partial exposure. Unlike prior approaches that perform one-shot decomposition and verification over the full input, DIVER periodically attempts to extract a verifiable atomic claim as the input unfolds. For each such claim, it generates a query, retrieves evidence through web search, and assesses whether the currently available information is sufficient to support a conclusion. If the evidence is insufficient, DIVER refines the query based on the retrieved results and continues iterative retrieval; if the evidence is sufficient, it produces a local SUPPORTED or REFUTED decision. Finally, once the full input has been revealed, DIVER performs a passage-level revision stage that revisits the complete text together with all previously verified claims, checks for missed errors, and produces a final verdict with a brief explanation.

We evaluate DIVER at two levels. First, it remains effective on standard one-shot fact-checking benchmarks. We then turn to the setting that most directly reflects our motivation: calibrated incremental fact-checking under partial exposure. To support this evaluation, Section 3.6 introduces UMACTUALLY, a machine-readable benchmark built from the Internet game *Um, Actually*. This task is challenging not only for computational systems, but also for humans. Using UMACTUALLY, we evaluate whether DIVER can not only identify factual errors, but also trigger at the right time, and we compare its behavior against human performance in the same section.

Contributions. Our contributions are as follows:

- We introduce DIVER, a new method for calibrated incremental fact-checking.
- We introduce UMACTUALLY, a machine-readable dataset built on the Internet game *Um, Actually*.
- We show that DIVER is effective on standard fact-checking benchmarks and supports direct comparison with human performance in a real-world incremental setting.

2 Method

This section presents DIVER, our LLM-based framework for calibrated incremental fact-checking. DIVER operates through an iterative control loop that periodically extracts verifiable atomic claims, retrieves and refines evidence through web search, and makes local support/refutation decisions when the available information is sufficient. After these incremental verification steps, DIVER performs a passage-level revision over the full input to recover missed errors, integrate previously verified claims, and produce a final verdict with a brief explanation. Figure 1 provides an overview of the framework.

Like many recent LLM-based fact verification pipelines, the main modules described below—including CLAIMEXTRACTOR, QUERYGENERATOR, FILTER, and FACTCHECKER—are imple-

mented as prompted LLM calls rather than separately trained models. Each module uses a task-specific prompt, and full prompting details are given in Appendix B.7.

2.1 Passage to Atomic Claims

The first stage of DIVER is **Claim Extractor**, which converts the revealed passage prefix into a set of verifiable atomic claims. Rather than decomposing the full passage in a single pass, Claim Extractor performs claim extraction incrementally as the input unfolds. At each extraction step, the model reads the current prefix and outputs one new atomic claim that can be checked independently, or returns a special STOP signal if no such claim is available.

An atomic claim in DIVER is a short factual statement that is specific enough to support retrieval and local verification. ClaimExtractor is prompted to preserve the meaning of the passage while rewriting it into a simpler, self-contained form suitable for downstream checking.

Consider the example in Figure 1. From the early prefix, ClaimExtractor may first extract a claim such as *Six Flags is named after its Texas roots*. After more context is revealed, it can produce a more specific claim such as *The name "Six Flags" refers to six nations that governed Texas*. Once the list is revealed, Claim Extractor can further generate claims such as *France governed Texas*, *Spain governed Texas*, and *England governed Texas*.

2.2 Iterative and Incremental Claim Verification Loop

As soon as Claim Extractor extracts a new atomic claim, DIVER immediately starts verifying it through an iterative retrieval-and-checking loop, while continuing to process other active claims in parallel. Unlike one-shot pipelines that retrieve once and immediately make a decision, DIVER continues retrieval for the same claim when the current evidence is still incomplete.

Query Generator (M_q). For each claim c_i, the query generator produces an initial search query

$$q_i = M_q(c_i). \tag{1}$$

Consider the claim from Figure 1: *The name "Six Flags" refers to the flags of the six nations that have governed Texas.* An initial query may be *meaning of Six Flags name* or *Six Flags name origin Texas*, which targets the central entities and relation in the claim.

Search Module ($\mathcal{R}$). The query is sent to the retriever $\mathcal{R}$, which returns a ranked list of candidate passages:

$$\mathcal{D}_i = \mathcal{R}(q_i). \tag{2}$$

For this example, the first retrieval round may return evidence stating that the name *Six Flags* comes from the phrase "six flags over Texas". This establishes one part of the claim, but does not yet identify the six governments themselves.

Filter & Query Recommendation (M_f). Given the claim c_i and the retrieved passages $\mathcal{D}_i$, the filter module selects the most relevant evidence and recommends a follow-up query:

$$(\mathcal{E}_i, q_i^{\text{rec}}) = M_f(\mathcal{D}_i, c_i). \tag{3}$$

In the running example, once the system has identified that the company name is tied to the "six flags over Texas," the remaining missing evidence is the list of those six governments. The recommended follow-up query may therefore be *what are the six flags over Texas* or *which six nations governed Texas*. The follow-up query is issued for the same claim, with the goal of completing the missing evidence chain.

Fact Checker (M_{ch}). The fact checker reads the claim together with the filtered evidence and outputs a verdict and a short rationale:

$$(v_i, \rho_i) = M_{ch}(c_i, \mathcal{E}_i), \tag{4}$$

where $v_i \in \{\text{SUP}, \text{REF}, \text{NEI}\}$.

In this example, the checker can make a decision only after both parts are available: evidence about what the name *Six Flags* refers to, and evidence about which six governments are meant. If one of these parts is still missing, the checker returns NOT ENOUGH INFORMATION, and DIVER continues retrieval using the recommended query. Otherwise, it assigns a final local label to the claim.

Iterative verification. This retrieve–filter–check loop continues until the claim is labeled SUPPORTED or REFUTED, or until a preset maximum number of retrieval rounds is reached. Verified claims are stored as tuples $\langle c_i, v_i, \rho_i \rangle$ and passed to the final revision stage. In this way, DIVER treats claim verification as a multi-step evidence accumulation process rather than a one-shot decision based on a single retrieval result.

2.3 Calibrated Buzzing and Revision

As discussed in Section 5, in the incremental fact-checking setting, the system is encouraged to make timely decisions rather than wait by default until the full passage has been observed. This timing behavior (Sung et al., 2025) is central to calibration: a well-calibrated verifier should avoid acting when the available evidence is still insufficient, but should trigger a fact check as soon as a claim is reliably refuted.

DIVER implements this through a simple buzzing policy. During the iterative claim verification loop, once any atomic claim is verified as REFUTED, the system immediately buzzes and outputs a paragraph-level verdict together with a brief explanation based on the refuted claim and its supporting evidence, while still continuing to process the remaining input and verify later claims until the passage ends.

Revision is invoked only when no such trigger occurs by the end of the passage. In this case, DIVER revisits the full passage together with the set of previously verified claims and asks whether an additional claim should be checked. The purpose of this stage is to recover errors that were missed during incremental verification, for example because they were not salient in any earlier prefix or only became clear after the full passage was observed.

Formally, at revision step t, the revision module proposes an additional claim

$$c_t^{\text{rev}} = M_r(P, \mathcal{C}_{t-1}), \tag{5}$$

where P is the full passage and $\mathcal{C}_{t-1}$ is the set of claims already verified before step t. The proposed revision claim is then sent through the same retrieve–filter–check loop described above. If it is verified as REFUTED, DIVER buzzes and outputs the final result; otherwise, revision continues until a preset revision budget is reached.

If no refuted claim is found even after revision, the system returns a final SUPPORTED decision with a brief explanation. In this way, DIVER combines early commitment when evidence is sufficient with a fallback recovery mechanism when incremental verification alone does not surface an error.

3 Experiments

This section evaluates DIVER from three perspectives. We first examine its effectiveness on standard public fact-checking benchmarks (Section 3.1–Section 3.3) to show that the framework is competitive under conventional one-shot evaluation. We then perform ablations and targeted analyses (Section 3.4– 3.5) to identify which components of DIVER drive these improvements. Finally, we study DIVER in a real-world incremental setting (Section 3.6) that directly tests its calibrated buzzing behavior, using UMACTUALLY, a new dataset built from the Internet game *Um, Actually*. Together, these experiments evaluate both the general effectiveness of DIVER and its ability to trigger fact checks at the right time under partial exposure.

3.1 One-Shot Fact Checking Comparison

Datasets. We evaluate on three widely used benchmarks chosen to cover different fact-checking settings, from short claim classification to retrieval-intensive open-web verification.

FEVEROUS-S (Aly et al., 2021) extends claim verification to evidence grounded in semi-structured Wikipedia pages, where systems must retrieve correct evidence and predict verdicts. **LIAR** (Wang, 2017) consists of short political claims with categorical labels, primarily stressing claim-level classification. **AVeriTeC** (Schlichtkrull et al., 2023) targets long-form, open-web fact verification with explicit evidence requirements and is evaluated with a composite score over evidence and verdict correctness.

Baselines. We compare against several representative categories: *(i) Zero-retrieval LLM*, a vanilla GPT model without external evidence; *(ii) LLM + web search*, where the model is prompted with retrieved web snippets; *(iii) Step-by-step / modular pipelines* that explicitly decompose verification into intermediate steps (HISS (Zhang and Gao, 2023), FactCheck-GPT (Wang et al., 2024), BiDeV (Liu et al., 2025)); and *(iv) Iterative retrieval-and-verification frameworks* that are closely related in spirit to DIVER (SAFE (Diao et al., 2024), FIRE (Xie et al., 2025), QACheck (Pan et al., 2023a), ProgRFC (Pan et al., 2023b)). For AVeriTeC, we additionally report results from strong *shared-task systems* like InFact (Rothermel et al., 2024) when available under the fixed-backbone setting (Table 2).

Evaluation Metrics. We follow the official evaluation protocols of each benchmark. On **FEVEROUS-S** and **LIAR** we report **Macro-F1**; on **AVeriTeC** we report the official **AVeriTeC score**, which jointly evaluates evidence retrieval and verdict correctness.

Implementation Details. For fair comparison, all LLM-based methods share the same open-web retrieval backend and decoding configuration unless otherwise specified. Here, *backbone* denotes the underlying LLM used by all prompted modules in a pipeline. We also compare several stopping policies for incremental claim extraction and revision, and use the best-performing configuration as the default in all experiments. Full implementation details, prompt templates, retrieval settings, Efficiency and cost statistics, and policy comparisons are provided in Appendix B.

3.2 Overall Performance

Table 1 reports the **backbone-swappable** comparison on FEVEROUS-S, LIAR, and AVeriTeC using two backbones. Across benchmarks, DIVER reports the strongest scores among pipeline baselines, with the largest margins on FEVEROUS-S and AVeriTeC, where retrieval quality and multi-hop evidence aggregation are especially important. Table 2 further fixes the backbone (GPT-4o) to enable direct comparison with competitive AVeriTeC shared-task systems under the standard evaluation protocol.

3.3 Main Results

Backbone-swappable results. DIVER achieves the best results on FEVEROUS-S and AVeriTeC under both backbones, and remains competitive on LIAR (Table 1). The improvement is larger on the weaker backbone (GPT-3.5-turbo), suggesting that DIVER's structured control—iterative extraction, evidence-aware query refinement, and paragraph-level revision—helps compensate for limited model capacity when evidence is sparse or noisy.

Comparison to iterative verification baselines. When compared against closely related iterative retrieval-and-verification frameworks (SAFE, FIRE, QACheck, ProgRFC), DIVER remains competitive on FEVEROUS-S and LIAR, indicating that its claim-centric control loop and revision stage provide complementary benefits beyond standard iterative querying. We further analyze which components contribute most in Section 3.4.

AVeriTeC shared-task comparison. Table 2 reports AVeriTeC results under a fixed GPT-4o backbone, allowing direct comparison to shared-task submissions (e.g., InFact, HerO). DIVER exceeds these systems under the official AVeriTeC score, demonstrating that the proposed pipeline remains strong even against task-specialized systems.

3.4 DIVER Benefits from Extraction, Retrieval, and Revision

We next analyze where DIVER's gains come from. We first use a leave-one-out ablation to measure the contribution of iterative extraction, adaptive retrieval, and revision, then study iterative extraction and adaptive follow-up retrieval more directly in targeted experiments.

We remove each major component in turn while keeping all other settings fixed. Table 3 reports the resulting scores.

Iterative extraction matters most on dense inputs. 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.

Adaptive retrieval gives the largest gains on multi-hop benchmarks. Without the filter/recommend module, 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 provides consistent gains. 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 3.

3.4.1 Iterative Extraction Exposes More Refutation-Critical Errors

We next test whether iterative extraction surfaces more refutation-critical errors than one-shot extraction. We evaluate the claim extractor as a standalone component using *coverage of gold incorrect sub-claims*, a task-aligned metric. Since paragraph-level verification is REFUTED once any incorrect sub-claim is identified, we ask whether the extractor can surface at least one such sub-claim for downstream retrieval and verification. Here, "coverage" means that the extracted claims include at least one semantically matching gold incorrect sub-claim, even if the wording differs.

Methods	FEVEROUS-S		LIAR		AVeriTeC score	
	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	0.17	0.12
LLM + Web Search	55.7	42.7	65.2	40.2	0.35	0.32
HISS (Zhang and Gao, 2023)	59.3	48.2	58.6	46.8	0.38	0.32
FactCheck-GPT (Wang et al., 2024)	65.3	56.5	65.2	52.9	0.50	0.44
BiDeV (Liu et al., 2025)	65.9	59.5	67.4	60.3	0.52	0.43
DIVER (ours)	**70.6**	**66.7**	**67.4**	**63.0**	**0.67**	**0.52**

Table 1: **Main results (backbone-swappable group).** For AVeriTeC, scores refer to the official metrics jointly evaluating evidence retrieval and verdict correctness.

Methods	FEVEROUS-S	LIAR Acc.	AVeriTeC score
SAFE (Diao et al., 2024)	55.1	62.6	N/A†
FIRE (Xie et al., 2025)	63.8	64.1	N/A†
QACheck (Pan et al., 2023a)	59.5	63.0	N/A†
ProgramFC (Pan et al., 2023b)	68.0	63.2	N/A†
AVeriTeC Shared Task Systems			
InFact (Rothermel et al., 2024)	66.2	67.0	0.63
HerO (HUMANE) (Yoon et al., 2024)	63.1	64.0	0.57
DIVER (ours, GPT-4o)	**70.6**	**67.4**	**0.67**

Table 2: **Main results (fixed-backbone group + AVeriTeC shared-task systems).** InFact and HerO AVeriTeC scores are official test-set numbers reported by the system papers / shared-task report. † N/A denotes methods that are not directly applicable to the AVeriTeC *score* setting (e.g., they do not produce the required evidence-format outputs for official scoring), hence no AVeriTeC score is reported.

Variant	FEV.	LIAR	AVT.
Full DIVER	**70.6**	**67.4**	**0.67**
– Iter. Extr.	65.3	66.8	0.56
– Filter / Rec.	61.2	63.1	0.58
– Revision	68.3	67.2	0.62

Table 3: Ablation results. Each row removes one major component from the full system while keeping all other settings fixed. All three components contribute, with the largest drops coming from iterative extraction and adaptive retrieval.

Setup. We sample 150 paragraph-length items from the REFUTED split of *AVeriTeC* (avg. 6.4 sub-claims per paragraph). For each paragraph we compare (i) a *one-shot* extractor that outputs all claims in a single pass and (ii) our *iterative* extractor that outputs one atomic claim per step until self-termination. Two annotators judge whether the extracted claim set contains *at least one* gold incorrect sub-claim (i.e., a refutation-critical error). We consider a predicted claim to cover a gold incorrect sub-claim if it is semantically equivalent to (or entails) that sub-claim. Annotation details are provided in Appendix C.

Metric. Let G^- be the set of annotated incorrect sub-claims for a paragraph and C be the extracted claim set. We define incorrect-subclaim coverage as $\text{ISC} = \mathbb{I}[\exists g \in G^-, \exists c \in C : \text{match}(c, g)]$, and report the average ISC over the sampled paragraphs.

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

Table 4: Incorrect-subclaim coverage (%) on 150 REFUTED AVeriTeC paragraphs: fraction of paragraphs whose extracted claim set contains at least one gold incorrect sub-claim.

Results and discussion. Iterative extraction improves incorrect-subclaim coverage by **+7.3 pp** on GPT-4o-mini and **+9.2 pp** on GPT-3.5-turbo (Table 4). The gain is more pronounced for the weaker backbone, suggesting that step-wise focusing helps lower-capacity models surface refutation-critical errors that a single-pass extractor often merges or omits.

3.4.2 Adaptive Follow-up Queries Improve Multi-hop Verification

To isolate the effect of dynamic follow-up 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*.Table 5 contrasts our results with representa-

Method	HoVer (hop-3)	HoVer (hop-4)
QACheck (Pan et al., 2023a)	54.67	52.35
FIRE (Xie et al., 2025)	57.23	55.12
FactCheck-GPT (Wang et al., 2024)	60.11	59.25
FLAN-T5 (Jiang et al., 2021)	60.23	55.42
BiDeV (Liu et al., 2025)	63.62	60.41
DIVER (ours)	**65.48**	**62.82**

Table 5: Accuracy (%) on HoVer hop-3 / hop-4 (open-domain). DIVER gives the best results among the compared systems, with larger gains as the hop count increases.

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 denotes failure to extract a refutation-critical sub-claim; RF denotes failure to retrieve sufficient evidence; UC denotes failure to identify a refutation despite sufficient evidence; SR denotes incorrectly labeling a supported claim as refuted; and Other covers errors outside these categories.

tive step-by-step baselines.

Findings DIVER improves over BiDeV, the best-performing baseline in Table 5, 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.

3.5 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).

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. For example, a paragraph state that *Marie Curie discovered polonium and later won the Nobel Prize in Chemistry*. A one-shot system may retrieve evidence only for the Nobel Prize and miss the earlier discovery-related sub-claim, whereas DIVER can continue verifying the remaining uncovered part through an additional retrieval step. 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. For instance, a paragraph says that *the film won three awards at a European festival*, the retriever may surface evidence about only one specific award event with a different count or scope, leading the checker to incorrectly treat this partial mismatch as a contradiction. 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.

3.6 UMACTUALLY: Real-World Calibrated Incremental Fact-Checking

Standard fact-checking benchmarks primarily evaluate whether a system can eventually produce the correct verdict once the full input is available. By contrast, the central question in our setting is not only *whether* a system identifies a factual error, but also *when* it decides to act under incremental exposure. To evaluate this directly, we introduce UMACTUALLY, a real-world benchmark built from the Internet game *Um, Actually*, where errors are embedded in dense spoken trivia and the input unfolds progressively. This setting directly matches the calibrated incremental fact-checking scenario motivating DIVER: the system must decide when the available evidence is sufficient to trigger a correction, rather than waiting by default until the full passage has been observed.

Dataset Construction. We extract UMACTUALLY from the 2024 and 2025 season of *Um, Actually*. Each example consists of a host statement containing at least one factual error together with the corresponding correction provided in the show. Two expert annotators segment each host statement into atomic claims and label these claims as SUP-

PORTED, REFUTED, or NOT ENOUGH INFORMATION, and additionally mark a *gold buzz position* for each paragraph. The resulting dataset contains **179** paragraphs, with an average of **6.8** atomic subclaims per paragraph. Additional details of UMACTUALLY are available in Appendix C

Unlike conventional benchmark construction, UMACTUALLY is designed to support both correctness and timing evaluation. The claim annotations allow us to measure whether the system identifies the relevant factual error, while the gold buzz positions allow us to measure whether it does so at an appropriate time as input unfolds.

Streaming Protocol and Buzz Position. To simulate the intended deployment setting, we reveal each paragraph *word by word* in subtitle order. At every step, the system may either continue reading or *buzz* and output a correction. For each example, annotators mark a gold buzz position $b^\star$, defined as the earliest word index after which an expert can correctly identify the relevant error using only the observed prefix.

When DIVER produces a correct correction at word index b, we measure its relative timing deviation as

$$\Delta = \frac{b - b^\star}{L} \times 100\%, \tag{6}$$

where L is the full paragraph length in words. This quantity normalizes timing across passages of different lengths and captures how far the system's buzz point lags behind gold buzz position. Lower values therefore indicate more timely decisions.

This protocol lets us evaluate calibrated buzzing directly. A strong system should not merely find the error eventually; it should buzz soon after enough evidence becomes available, while avoiding premature triggers based on incomplete context.

End-to-End Results. To the best of our knowledge, DIVER is the only system in our comparison that is natively designed for this calibrated incremental setting. However, existing fact-checking pipelines such as BiDeV can still be adapted to this setting through repeated prompting over progressively revealed input, making direct comparison possible. Table 7 compares DIVER against two strong baselines that share the same Google Search retriever and GPT-4o backbone. DIVER reports the best results on UMACTUALLY, improving over BiDeV by **+10.09** Macro-F1 and **+9.79** Accuracy. These gains are consistent with the trends observed on other retrieval-intensive benchmarks, but are especially important here because correctness is evaluated in a noisy incremental setting rather than under full-input access.

Method	Macro-F1 (mean ± 95% CI)	Accuracy (mean ± 95% CI)
Vanilla GPT-4o	34.48 ± 3.17	33.70 ± 3.01
BiDeV	70.39 ± 8.03	71.85 ± 6.66
DIVER (ours)	$\mathbf{81.28 \pm 6.04}$	$\mathbf{81.64 \pm 5.19}$

Table 7: Results on UMACTUALLY over five independent runs, reported as the mean and 95% confidence interval. DIVER remains the strongest system under the multi-run evaluation.

These results show that DIVER remains effective even when the task is no longer simply to verify a static paragraph after complete observation. Instead, it must operate under progressively revealed input and commit at the right time. In this regime, the gains from iterative extraction, adaptive retrieval, and calibrated buzzing translate into stronger end-to-end fact-checking quality.

Buzz Timing. Correctness alone is not sufficient in this setting; the more distinctive question is whether DIVER *buzzes at the right time*. Using the timing metric above, we examine the distribution of Δ over correctly answered examples. DIVER buzzes within **20%** of the gold earliest position for **52.5%** of cases, and within **40%** for **78.7%** of cases. This indicates that, when DIVER is correct, it typically identifies the relevant error shortly after enough evidence becomes available.

To better characterize the limitations of DIVER's buzzing behavior, we analyze the buzzing timing on the 37 examples it fails in the same run. DIVER produces an incorrect buzz in 30 cases: 21 occur before the gold buzz position and 9 occur after it. In the remaining 7 cases, DIVER never buzzes. Thus, DIVER's primary timing failure mode is premature commitment rather than late incorrect buzzing. This finding is consistent with the relatively high frequency of spurious-refutation errors in Table 6 and highlights the importance of improving calibration against premature buzzing.

Human Comparison. Because UMACTUALLY is derived from a human-played quiz format, it also provides a natural reference point for human incremental fact-checking. We report show-level human accuracy using the earliest correct contestant answer for each question when multiple answers are attempted. Across **179** questions from the 2024 and 2025 seasons, contestants answered **130** cor-

rectly and **49** incorrectly, for an overall accuracy of **72.6%**. Accuracy varies across seasons: the 2025 season reaches **79.8%** (95/119), whereas the 2024 season reaches **58.3%** (35/60).

We also compare buzz timing on the subset of 2025-season questions answered correctly by humans. Using the same timing metric in Equation 6, the average human timing deviation is **26.7%** over these **95** questions. DIVER is slightly better on this metric, suggesting that its gains are not limited to final correctness, but also extend to timely triggering under incremental exposure.

4 Conclusion

We presented DIVER, an LLM-based framework for calibrated incremental fact-checking. Unlike prior pipelines that assume full-input access and one-shot verification, DIVER verifies a passage under progressively revealed input: it incrementally extracts atomic claims, iteratively retrieves and refines evidence, and buzzes as soon as a claim is reliably refuted, with a revision stage as fallback when no such trigger occurs. Across FEVEROUS-S, LIAR, and AVeriTeC, DIVER reports stronger results than competitive baselines, and our analyses show that these gains come from the intended mechanisms: iterative extraction, adaptive retrieval, and passage-level revision. To evaluate the central question of *when* a system should act, we introduced UMACTUALLY, a real-world benchmark derived from *Um, Actually*. Results on UMACTUALLY show that DIVER not only improves fact-checking quality, but also usually triggers close to the earliest valid decision point.

With advances in model capabilities and system engineering, timely fact-checking is receiving growing attention. Real-world scenarios such as social media reply chains, spoken question answering, and live commentary require reliable verification before the full input is available. We believe our work reflects where the field stands on this emerging problem and lays the groundwork for better systems, broader evaluation, and real-world deployment. We hope it inspires broader interest and further work in this direction, including multimodal incremental verification, greater robustness to adversarial examples and misleading partial evidence, and better buzz thresholds and timing policies that adapt to uncertainty and application needs.

5 Related Work

Benchmarks beyond simple claims. Early fact-checking benchmarks such as FEVER (Thorne et al., 2018) and LIAR (Wang, 2017) enabled large-scale evaluation, but their inputs are often short and simple, understating the difficulty of realistic, compositional statements. Subsequent datasets increase complexity in complementary ways: Fool Me Twice (Eisenschlos et al., 2021) emphasizes compositional and adversarial claims; FEVEROUS (Aly et al., 2021) and FEVEROUS-S add semi-structured evidence such as tables and lists; AVeriTeC (Schlichtkrull et al., 2023) targets paragraph-level, open-domain verification that often requires multi-hop evidence; and SciFact (Wadden et al., 2020) introduces scientific evidence and distribution shift. Together, they expose a key challenge: identifying refutation-critical information embedded in dense context, not merely predicting labels.

Incremental answering and calibration. Our setting is also related to incremental QA and calibration. Quizbowl-style QA studies when a system should answer under progressively revealed input, framing buzzing as a sequential decision tied to confidence calibration (Rodriguez et al., 2019). GRACE similarly evaluates how early, accurately, and confidently models answer as evidence is revealed (Sung et al., 2025). These works motivate our view that correctness alone is insufficient under incremental exposure: a system should defer when evidence is weak but commit once it becomes reliable.

Decomposition and agentic verification pipelines. Prior work handles complex inputs through sentence decomposition (Liu et al., 2020), claim segmentation (Chen et al., 2022), and structured multi-step frameworks such as ProgramFC and QACheck (Pan et al., 2023b,a). LLM-based pipelines further interleave retrieval, reasoning, and evidence judging (Lewis et al., 2020; Wei et al., 2022; Press et al., 2023; Xie et al., 2025; Diao et al., 2024). However, they typically assume either full-input access or a fixed decomposition before verification. In contrast, DIVER targets incremental, calibration-sensitive fact-checking: it repeatedly revises what to check, refines retrieval when evidence remains insufficient, and explicitly decides when to buzz rather than waiting for the full passage.

Limitations

Single–run evaluation. All metrics reported in Tables 2–3 originate from a *single inference run* per system. Because OpenAI API usage incurs a non-trivial cost ($100 total; Appendix B.5), we did not perform multiple seeds or temperature sweeps. Future work should measure variance across runs.

Model dependence on proprietary LLMs. Our system relies on `gpt-3.5-turbo-0125` and `gpt-4o-2024-07-18`, whose weights are not publicly available. Re-training or fine-tuning open-source models may reduce cost and improve reproducibility.

Dataset coverage. AVeriTeC, FEVEROUS, and LIAR cover political and Wikipedia-style claims. Performance on domains such as medical or financial fact-checking has not been tested and may vary.

Human annotation scale. We used two adult volunteers for claim-quality inspection. While sufficient for the pilot study, larger and more diverse annotator pools are required to robustly measure inter-annotator agreement.

Use of AI assistants. The authors used ChatGPT only for limited language polishing of earlier manuscript drafts. No AI assistant was used to design the method, run experiments, interpret results, or make scientific decisions. All technical content and claims in the final paper were produced and verified by the authors.

Potential Risks

Like other automated fact-checking systems, DIVER may be misused or over-trusted in high-stakes settings. In particular, an incorrect SUPPORTED judgment may reinforce a false statement, while an incorrect REFUTED judgment may unfairly cast doubt on a true one. This risk is especially important in our incremental setting, where the system is encouraged to act before the full passage has been observed: a poorly calibrated model may buzz too early based on incomplete or misleading partial evidence.

Our system also relies on external retrieval and large language models, which introduces additional risks. Retrieved evidence may be incomplete, irrelevant, or misleading, and downstream model errors may propagate into the final decision. Because DIVER makes timing-sensitive decisions, such errors can affect not only whether a prediction is correct, but also when the system chooses to intervene.

We therefore do not view DIVER as a replacement for expert human judgment, especially in sensitive domains such as health, law, or politics. Instead, we envision it as a decision-support tool used with human oversight. Future work should further improve calibration, robustness to adversarial or misleading evidence, and uncertainty-aware abstention before deployment in real-world settings.

Acknowledgments

We thank the creators and production team of *Um, Actually* at College Humor, particularly Mike Trapp, for providing the source material used to construct the UMACTUALLY dataset and format. We also thank members of the University of Maryland CLIP Lab for their helpful discussions and feedback. Boyd-Graber's research is supported by a gift from the Adobe Corporation. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsors.

References

Rami Aly, Zhijiang Guo, Michael Sejr Schlichtkrull, James Thorne, Andreas Vlachos, Christos Christodoulopoulos, Oana Cocarascu, and Arpit Mittal. 2021. FEVEROUS: Fact extraction and VERification over unstructured and structured information. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*.

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.

Shizhe Diao, Pengcheng Wang, Yong Lin, Rui Pan, Xiang Liu, and Tong Zhang. 2024. Active prompting with chain-of-thought for large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1330–1350, 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, Online. Association for Computational Linguistics.

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 *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

Yuxuan Liu, Hongda Sun, Wenya Guo, Xinyan Xiao, Cunli Mao, Zhengtao Yu, and Rui Yan. 2025. Bidev: bilateral defusing verification for complex claim fact-checking. In *Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence and Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence and Fifteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'25/IAAI'25/EAAI'25. AAAI Press.

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 *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6996–7045, Vienna, Austria. Association for Computational Linguistics.

Liangming Pan, Xinyuan Lu, Min-Yen Kan, and Preslav Nakov. 2023a. QACheck: A demonstration system for question-guided multi-hop fact-checking. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 264–273, Singapore. Association for Computational Linguistics.

Liangming Pan, Xiaobao Wu, Xinyuan Lu, Anh Tuan Luu, William Yang Wang, Min-Yen Kan, and Preslav Nakov. 2023b. Fact-checking complex claims with program-guided reasoning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6981–7004, Toronto, Canada. Association for Computational Linguistics.

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. *arXiv preprint arXiv:1904.04792*.

Mark Rothermel, Tobias Braun, Marcus Rohrbach, and Anna Rohrbach. 2024. InFact: A strong baseline for automated fact-checking. In *Proceedings of the Seventh Fact Extraction and VERification Workshop (FEVER)*, pages 108–112, Miami, Florida, USA. Association for Computational Linguistics.

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*, volume 36, pages 65128–65167. Curran Associates, Inc.

Sandeep Singamsetty, Nishtha Madaan, Sameep Mehta, Varad Bhatnagar, and Pushpak Bhattacharyya. 2023. "Beware of Deception": Detecting Half-Truth and Debunking It through Controlled Claim Editing. *arXiv preprint arXiv:2308.07973*.

Yoo Yeon Sung, Eve Fleisig, Yu Hou, Ishan Upadhyay, and Jordan Boyd-Graber. 2025. GRACE: A granular benchmark for evaluating model calibration against human calibration. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 19586–19587, Vienna, Austria. Association for Computational Linguistics.

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.

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. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 14199–14230, Miami, Florida, USA. Association for Computational Linguistics.

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.

Zhuohan Xie, Rui Xing, Yuxia Wang, Jiahui Geng, Hasan Iqbal, Dhruv Sahnan, Iryna Gurevych, and Preslav Nakov. 2025. FIRE: Fact-checking with iterative retrieval and verification. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 2901–2914, Albuquerque, New Mexico. Association for Computational Linguistics.

Yejun Yoon, Jaeyoon Jung, Seunghyun Yoon, and Kunwoo Park. 2024. HerO at AVeriTeC: The herd of open large language models for verifying real-world claims. In *Proceedings of the Seventh Fact Extraction and VERification Workshop (FEVER)*, pages 130–136, Miami, Florida, USA. Association for Computational Linguistics.

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 the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 996–1011, Nusa Dua, Bali. Association for Computational Linguistics.

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.

A Additional Dataset UMACTUALLY Details

A.1 Annotation protocol.

Annotators first decompose each host statement into self-contained atomic claims. Each extracted claim must be independently interpretable: annotators resolve pronouns, supply necessary context, and rewrite the claim so that it can be verified without relying on the surrounding passage. For claims corresponding to the refuted statement already identified in the original *Um, Actually* source, annotators record the correction. For all other extracted claims, annotators independently verify the claim, assign a label from SUPPORTED, REFUTED, NOT ENOUGH INFORMATION, and provide a brief explanation. For each REFUTED claim, annotators also mark a gold buzz position, defined as the earliest word position at which the refuted claim can be extracted using only the observed prefix.

Because claim extraction is open-ended, exact agreement over extracted claim sets is not straightforward to compute. Instead, the authors inspected both annotators' outputs and returned high-disagreement cases for discussion. These included cases where annotators extracted substantially different numbers of claims, identified different refuted claims, or assigned different gold buzz positions. Final annotations were produced after discussion and adjudication.

A representative annotated example corresponding to Figure 1 is shown in Table 8.

B Detailed Experimental Settings

B.1 Data and Pre-processing

Datasets. We evaluate on the **dev** split of three public fact-checking corpora:

- **AVeriTeC**[2]
- **LIAR**[3]
- **FEVEROUS**[4]

We treat the task as *open-domain* retrieval and therefore ignore the official evidence fields and the provided Wikipedia corpora. No prompt or

[2] https://fever.ai/dataset/averitec.html
[3] https://zenodo.org/records/14011838
[4] https://fever.ai/dataset/feverous.html

Claim	Label	Correction	Buzz position
Six Flags might have parks all over the country.	SUPPORTED	—	—
Six Flags is named after its Texas roots.	SUPPORTED	—	—
Six Flags' name refers to the flags of the six nations that have governed Texas.	SUPPORTED	—	—
France governed Texas.	SUPPORTED	—	—
Spain governed Texas.	SUPPORTED	—	—
England governed Texas.	REFUTED	England did not govern Texas; the Confederacy, not England, is the sixth flag.	32
Mexico governed Texas.	SUPPORTED	—	—
The Republic of Texas governed Texas.	SUPPORTED	—	—
The United States governed Texas.	SUPPORTED	—	—

Table 8: Annotations for the representative example shown in Figure 1.

hyper-parameter tuning is performed on the train split.

Label filtering.

- For **AVeriTeC** and **FEVEROUS** we retain only claims whose gold label is `SUPPORTED` or `REFUTED`, discarding `NOT ENOUGH INFO`.
- Following prior work (e.g. (Singamsetty et al., 2023)), **LIAR**'s six-way labels are mapped to binary: `pants-fire` & `false` → `REFUTED`; `mostly-true` & `true` → `SUPPORTED`; `half-true` & `barely-true` are discarded.

Dataset	# SUPPORTED	# REFUTED
AVeriTeC	122	305
LIAR	449	341
FEVEROUS	3 908	3 481

Table 9: Claim counts after filtering.

Randomness. Pre-processing is deterministic; no sampling or shuffling is involved, thus no random seed is required.

B.2 API Configuration

All methods (HISS, FactCheck-GPT, BiDeV, **DIVER**) call **OpenAI** end-points:

- `gpt-3.5-turbo-0125`: `temperature`=0.3, `top_p`=0.95
- `gpt-4o-mini-2024-07-18`: default decoding parameters

Other parameters are left at their SDK defaults (`max_tokens`=none, `n`=1, `stop`=nil, `presence_penalty`=0, `frequency_penalty`=0).

Web search is performed via **SERPER** API (`gl=us`, `hl=en`, `num=10`). All prompts are included in the code provided in the supplementary materials.

B.3 Baseline Re-implementation

For HISS, FactCheck-GPT and BiDeV we use the original repositories *unaltered* except for:

- replacing the search module with SERPER (identical `top-k`=10)
- injecting our API key and locking model names to the snapshots above

B.4 Hardware & Software Environment

Experiments run on a **Windows 11** workstation (AMD Ryzen 7 8845H, 32 GB RAM, no GPU). Key libraries: `python 3.11.11`, `openai 1.66.3`, `tiktoken 0.6.0`, `scikit-learn 1.5.0`.

B.5 Cost & Efficiency

The total runtime of our experiments was less than 20 CPU-hours, with approximately $100 in OpenAI API usage. We further compare the inference cost of DIVER with SAFE and FIRE. However, DIVER operates under a different setting: its cost consists of both (i) the verification process itself and (ii) the additional overhead required for incremental processing. SAFE and FIRE verify fully observed claims, whereas DIVER processes input incrementally and must repeatedly determine whether the newly revealed prefix contains a verifiable claim before retrieval and verification begin.

DIVER's verification process itself has a cost comparable to FIRE: it uses fewer LLM calls (8.4 vs. 9.6), with only modest increases in searches (+0.6) and token usage (+199). The remaining cost comes from incremental claim monitoring, which adds 5.1 lightweight LLM calls and 1,674 tokens per input on average, but requires no additional searches. This overhead reflects the difference in task setting: SAFE and FIRE assume that the full claim or passage is available before verification begins, whereas DIVER targets settings in which input is revealed progressively, such

Method / Component	Avg. LLM calls	Avg. searches	Avg. tokens
SAFE	6.2	2.8	7,672
FIRE	9.6	4.4	12,203
DIVER verification process	8.4	5.0	12,402
DIVER incremental overhead	5.1	0.0	1,674
DIVER total	**13.5**	**5.0**	**14,076**

Table 10: Average inference cost per input. DIVER's cost is decomposed into the verification process shared with conventional fact-checking pipelines and the additional overhead required for incremental claim monitoring.

as live commentary, spoken question answering, and unfolding social-media discourse. In these settings, the system must determine not only whether a claim is true, but also whether the currently observed prefix contains a claim that is ready for verification.

B.6 Iteration–Policy Ablation

Both the claim-extraction loop (i iterations per sentence) and the revision loop (j iterations per paragraph) can be governed by different stopping policies. We evaluate four alternatives for i and three for j.

Claim extraction policies (i).

- **CE-1** *Self-termination.* The extractor outputs a special STOP token once it believes no novel atomic claim remains.
- **CE-2** *Fixed budget.* We run a fixed number of steps $i = n$; dev tuning selects n=5.
- **CE-3** *Length-based.* $i = \lceil |u|/n \rceil$ with n=4 tokens, allocating more steps to longer sentences.
- **CE-4** *Entity-based.* $i = n \times \#\text{NE}(u)$ with n=3, where $\#\text{NE}(u)$ is the number of named entities in u.

Revision policies (j).

- **REV-1** *Fixed budget.* We run a fixed number of revision passes $j = n$ with n=4.
- **REV-2** *Slack budget.* $j = \max\big(n - |\mathcal{C}|,\, 0\big)$ with n=15, decreasing the remaining budget as more claims are extracted.
- **REV-3** *Length–entity hybrid.* $j = \max\big(\lceil |P|/n \rceil - d\,|\mathcal{C}|,\, 0\big)$, where n=4 and per-claim discount d=1.

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). 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 11) offers the best balance between recall and precision without incurring excessive cost.

B.7 Prompt Templates

This appendix reports the prompt templates used by the main modules in DIVER. All modules are implemented as prompt-based LLM calls.

ClaimExtractor.

> You are a fact-checking assistant. Given a sentence and a list of key points that have already been identified (can be empty) for fact-checking, your task is to find one (only one!) additional key point from the sentence that is not already in the list and requires fact-checking.
>
> Return only the new key point. Do not include any explanation or extra text.
>
> If you think all the key points are already identified, simply reply with NULL.

FactChecker.

> You are a fact-checking assistant. Your job is to determine whether a given claim is factually correct, based only on the evidence provided. You must output one of the following options:
>
> "Correct" — if the evidence clearly supports the claim.
>
> "Incorrect. (followed by a brief explanation)" — if the evidence contradicts the claim.
>
> "Not enough information" — if the evidence is insufficient or ambiguous.
>
> Be concise and only rely on the provided evidence. Do not use prior knowledge or make assumptions beyond what is stated.

Claim Extractor	Revision Stage	AVeriTeC	FEVEROUS
CE-1	REV-1	**83.0**	**70.6**
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

Table 11: Accuracy on AVeriTeC and FEVEROUS dev for representative iteration-policy pairs. CE-1 + REV-1 is selected as the default configuration.

If the claim contains emotional or exaggerated language, or if it is a statement that is clearly an opinion or rhetorical flourish (such as "I'm smart" or "I know how to game the system"), or if the claim is somehow None or unrecognizable, output "Correct" as these types of claims do not require fact-checking.

QueryGenerator.

You are a search query expert. Your job is to convert factual claims into clear, concise, and effective search engine queries in English. Your queries should help retrieve evidence that confirms or refutes the claim.

Always output the query in plain text. Do not include explanations or formatting.

Filter.

You are assisting in verifying the truth of a factual claim.

Below you will find (1) the claim and (2) a passage taken from a webpage.

Your task

1. Copy **exactly** any sentence(s) or paragraph(s) from the passage that provide evidence for deciding whether the claim is true or false.

2. If you notice text that looks potentially helpful but would still require further Google searching or multi-hop reasoning to become decisive, copy that text as well.

3. When you include such "potential evidence", add one extra line at the very end of your answer in the following format:

Recommended next query: <a concise Google search query that would likely surface the needed follow-up evidence>

4. Do **not** rewrite, summarize, or reorder any text you copy.

5. If the passage contains no relevant information, return exactly:

No relevant information found.

Revisor.

You are a fact-checking assistant. Given a paragraph and a list of key points that have already been identified for fact-checking, your task is to find one (only one!) additional key point from the paragraph that is not already in the list and requires fact-checking.

Return only the new key point. Do not include any explanation or extra text.

C Ethics & Licensing

Data licensing & privacy. AVeriTeC and FEVEROUS contain Wikipedia text licensed under CC-BY-SA 3.0; LIAR consists of public political statements. We redistribute only claim texts and binary labels, with no personal or sensitive information.

Human subjects & IRB. Our annotation task involved two adult student volunteers from our research labs, who annotated *publicly available* claims. The annotators were informed that the resulting annotations and derived dataset would be used for research and released as part of this project. We did not collect demographic or personally identifying information beyond their voluntary participation in the annotation process, and the study did not involve private or sensitive user data. All annotators provided informed consent before participation.

Model usage compliance. All calls to the OpenAI API comply with its Terms of Use. No user-generated private data was transmitted.

Societal impact. While automated fact-checking helps curb misinformation, erroneous "SUPPORTED" judgements may reinforce falsehoods. We recommend human-in-the-loop deployment, especially in high-stakes domains (e.g. medical).

Environmental impact. API inference consumed < 0.5 kg CO_2-e (0.04 kg/kWh, 12 kWh total).