The Missing Parts: Augmenting Fact Verification with Half Truth Detection

Yixuan Tang　　**Jincheng Wang**　　**Anthony K.H. Tung**

School of Computing, National University of Singapore

yixuan@comp.nus.edu.sg, bertrand.wongjc@gmail.com, atung@comp.nus.edu.sg

Abstract

Fact verification systems typically assess whether a claim is supported by retrieved evidence, assuming that truthfulness depends solely on what is stated. However, many real-world claims are *half-truths*, factually correct yet misleading due to the omission of critical context. Existing models struggle with such cases, as they are not designed to reason about omitted information. We introduce the task of **half-truth detection**, and propose POLITIFACT-HIDDEN, a new benchmark with 15k political claims annotated with sentence-level evidence alignment and inferred claim intent. To address this challenge, we present **TRACER**, a modular re-assessment framework that identifies omission-based misinformation by aligning evidence, inferring implied intent, and estimating the causal impact of hidden content. TRACER can be integrated into existing fact-checking pipelines and consistently improves performance across multiple strong baselines. Notably, it boosts Half-True classification F1 by up to 16 points, highlighting the importance of modeling omissions for trustworthy fact verification. The benchmark and code are available via https://github.com/tangyixuan/TRACER.

1 Introduction

The rapid spread of digital content has made fact verification a critical component in combating misinformation and promoting trustworthy public discourse. Traditional fact-checking systems follow a standard paradigm: given a claim and a body of evidence, the system classifies the claim as *true*, *false*, or *not enough information* (Chen and Shu, 2024). These systems are effective in identifying clearly incorrect claims and continue to serve as the backbone of automated verification pipelines.

However, many real-world claims are not outright false but are still misleading due to the omission of critical context. Misinformation can evolve

---

Claim: Under our administration, unemployment has fallen to its lowest level in half a century, demonstrating that our economic policies are working.

Presented Evidence (PE):
• Official labor statistics confirm the unemployment rate dropped to 3.5%, the lowest in 50 years.

Hidden Evidence (HE):
• Most of the new jobs were part-time or gig-based, lacking benefits or job security. → CHE
• Labor force participation remained low, with many discouraged workers no longer counted. → CHE
• Job growth was particularly strong in the hospitality and retail sectors.

Verdict by Standard FV Model:
True: The claim is factually supported by official statistics.

TRACER Re-Assessment Verdict:
Half-True: Although the unemployment figure is accurate, the omission of job quality and participation context distorts the implied economic success.

Table 1: A factually correct political claim re-evaluated as misleading (Half-True) by TRACER through Critical Hidden Evidence (CHE) analysis.

---

dynamically when propagated under different political stances (Chong et al., 2025), these are often referred to as **half-truths**, i.e. statements that are factually correct but strategically incomplete (Singamsetty et al., 2023; Jaradat et al., 2024). Consider the example in Table 1, where a politician claims that unemployment has reached a 50-year low. While this statistic is factually accurate, it omits key information, such as the rise in part-time gig jobs and stagnant labor force participation, that undermines the implied narrative of broad economic success. Standard fact verification (FV) models, which focus on validating surface-level factuality, label such claims as *true*, failing to capture the misleading nature of selective omission.

This challenge highlights a fundamental limitation in existing FV pipelines: they are not designed to reason about what is missing. Current models typically assess what is stated, treating veracity as a discrete property grounded in textual entail-
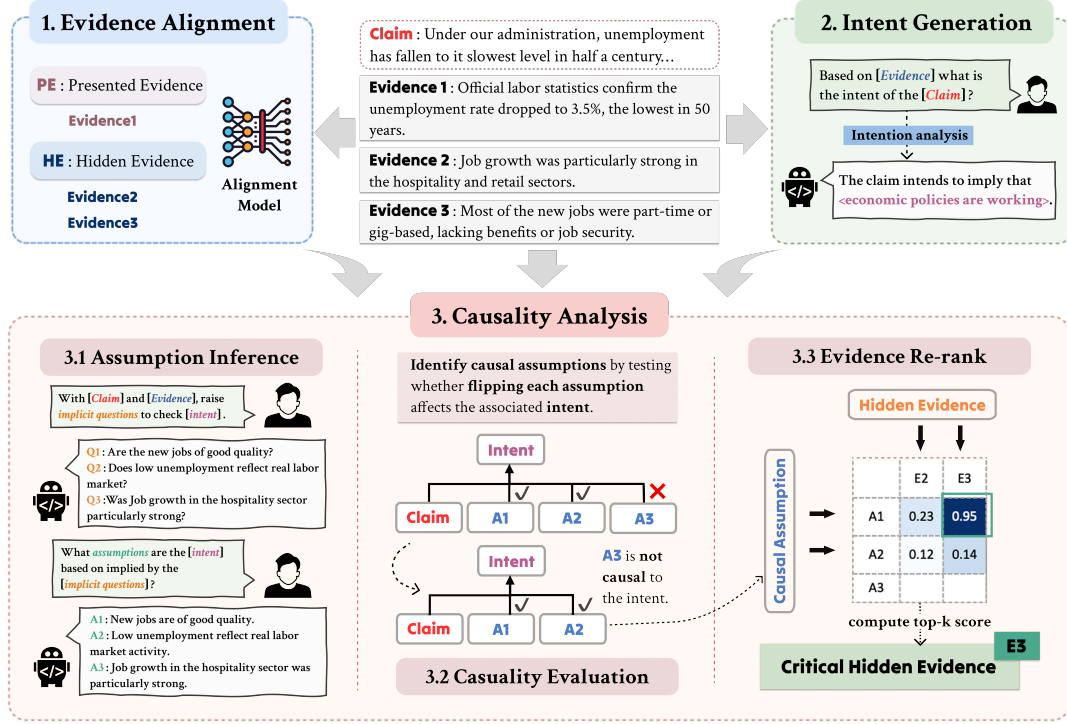
Figure 1: Overview of the TRACER framework for half-truth detection. The system identifies Critical Hidden Evidence (CHE) through evidence alignment, intent generation, and causality analysis, and re-assesses claims for omission-based misinformation.

ment (Molina et al., 2019; Estornell et al., 2020). Yet in practice, truthfulness is often shaped by both what is said and what is left unsaid. Omission-based misinformation exploits this gap, occupying a gray area between truth and falsehood that standard systems are ill-equipped to address.

In this paper, we introduce the task of **half-truth detection**, which complements traditional fact verification by modeling *completeness*. We define half-truths as claims that are factually accurate but omit **Critical Hidden Evidence (CHE)**—information that, if included, would significantly alter the plausibility of the claim's implied meaning. Our goal is to identify such omissions and assess their impact on the inferred intent of the claim.

To tackle this task, we propose **TRACER** (*Truth ReAssessment with Critical Hidden Evidence reasoning*), a framework to augment fact-checking systems with omission-aware reasoning. TRACER operates in three stages: (1) **evidence alignment**, to classify retrieved evidence as presented or hidden; (2) **intent generation**, to recover the claim's implicit message; and (3) **causality analysis**, to determine whether the Hidden Evidence undermines the inferred intent. These components feed into a lightweight **re-assessment module** that revisits claims, particularly those initially labeled as *true*,

and identifies misleading omissions. TRACER is model-agnostic and can be integrated into both agent-based and prompting-based FV pipelines.

To support this task, we construct **POLITIFACT-HIDDEN**, a benchmark dataset based on the Politi-Fact corpus. It contains about 15k claims annotated with sentence-level labels indicating Presented and Hidden Evidence, along with inferred claim intents validated through a combination of LLM prompting and human quality control. To our knowledge, this is the first dataset to explicitly annotate both omission and intent, enabling systematic study of half-truths at scale.

Our contributions are as follows:

1. We **formulate half-truth detection** as a new task in fact verification, targeting claims that omit critical context while remaining factually correct.

2. We introduce **POLITIFACT-HIDDEN**, a large-scale benchmark with fine-grained annotations for Presented / Hidden Evidence and inferred claim intent.

3. We propose **TRACER**, a three-stage framework that identifies omission-based misinformation through evidence alignment, intent

modeling, and causal reasoning. TRACER can be deployed as a re-assessment module and yields substantial gains in detecting half-truths across multiple strong baselines.

By modeling completeness alongside correctness, this work advances the frontier of fact verification. It addresses a blind spot in current systems and offers a generalizable framework for uncovering more subtle forms of misinformation that operate through omission rather than distortion.

2 Related Work

Fact Verification. Fact verification is commonly framed as a three-stage pipeline involving claim detection, evidence retrieval, and claim classification into *Supported*, *Refuted*, or *Not Enough Information* (Thorne et al., 2018; Guo et al., 2022). Benchmarks such as FEVER (Thorne et al., 2018) and LIAR (Wang, 2017) have facilitated significant progress in this area. Most existing systems focus on surface-level factual correctness, aiming to match claims against retrieved facts. While effective for outright falsehoods, these approaches are less suited to handling omission-driven manipulation.

Omission and Half-Truths. Omission-based misinformation, including half-truths, has received increasing attention. Singamsetty et al. (2023) introduce controlled claim editing to expose omitted content, and Chen et al. (2022) propose generating implicit questions to recover missing context. Other datasets have incorporated related annotations, such as *Cherry-picking* (Schlichtkrull et al., 2023) and *Mixture* (Yang et al., 2022), which primarily capture conflicting evidence rather than omissions per se. These schemes focus on factual inconsistency (i.e., presence of both supporting and refuting evidence), rather than semantic incompleteness or intent-driven distortion. In contrast, our work targets *half-truths*, claims that are factually accurate but strategically omit Critical Hidden Evidence (CHE) that significantly alters interpretation. Closely related are efforts that explore the role of intent in misinformation, such as distinguishing disinformation through concealment and overstatement (Rodríguez-Ferrándiz, 2023; Lee and Lee, 2024). Tang et al. (2025) uncover the comprehensive view of events by mitigating selective presentation of information, they do not integrate downstream fact verification. We go beyond these by

explicitly modeling the causal impact of Hidden Evidence on inferred intent without altering the original claim.

Reasoning-Based Fact Checking. Recent methods incorporate structured reasoning to improve factuality assessment. Program-guided models such as QACheck and ProgramFC (Pan et al., 2023b) generate intermediate steps to support verification (Tang et al., 2021). Argumentation-based approaches, such as CHECKWHY (Si et al., 2024), model causal links within evidence chains. Meanwhile, prompting-based methods like HiSS (Zhang and Gao, 2023) and Flan-T5 (Chung et al., 2022) leverage large language models for step-by-step verification. Other work explores intent modeling using contrastive learning (Yang et al., 2024) or refined retrieval (Wang et al., 2024). Our work complements these efforts by introducing omission-aware reasoning and providing a modular framework that can be integrated into both structured and generative pipelines.

3 Task Formulation

We define **half-truth detection** as an extension of fact verification that focuses on *factual completeness*. A claim may be factually accurate in isolation, yet convey a misleading impression by omitting relevant information that influences its interpretation. The goal is to identify such omissions and assess whether they materially affect the plausibility of the claim's implied message.

Formally, given a claim C and a set of retrieved evidence sentences $E = \{e_1, e_2, \ldots, e_n\}$ relevant to C, the goal is to classify the claim into one of three categories: *True*, *Half-True*, or *False*. This classification is determined not only by factual support but also by the presence or absence of **Critical Hidden Evidence (CHE)** $\subseteq E$ that is both (1) not presented in the claim, and (2) necessary to understand or challenge the claim's implied conclusion.

To support this, we define the following components:

- **Presented Evidence (PE)**: Sentences in E that are explicitly stated or clearly implied in the claim.

- **Hidden Evidence (HE)**: Sentences in E that are relevant to the claim but not mentioned.

- **Intent**: The implied conclusion or message that the claim is likely to convey to the reader.

Consolidated Label	Original Rating(s)
True	True
Half-True	Mostly True, Half-True
False	Mostly False, False, Pants on Fire

Table 2: Mapping from original PolitiFact ratings to consolidated labels.

Split	True	Half-True	False	Total
Train	1,352	4,564	6,078	11,994
Dev	64	195	741	1,000
Test	93	406	1,501	2,000

Table 3: Distribution of labels in the POLITIFACT-HIDDEN dataset across train/dev/test splits.

- **Critical Hidden Evidence (CHE)**: A subset of HE that, if revealed, would significantly affect the plausibility of the claim's intent.

This formulation connects closely to the traditional FV pipeline but adds a new layer of reasoning: not only must a system verify what is said, it must also reason about what is left unsaid. By focusing on omissions that shift the meaning of a claim, half-truth detection supports a more nuanced understanding of misinformation and helps uncover subtle forms of manipulation that standard FV systems may overlook.

4 Dataset: POLITIFACT-HIDDEN

As illustrated in Figure 2, we develop a semi-automated annotation pipeline (Figure 2) combining `GPT-4o-mini` prompting and model-assisted refinement to label each claim with evidence alignment and Intent.

We introduce POLITIFACT-HIDDEN, a benchmark for omission-aware fact verification. It extends the original PolitiFact corpus with fine-grained annotations capturing both Presented and Hidden Evidence, and the Intent behind each claim. These annotations enable systematic evaluation of whether omitted content, i.e. Critical Hidden Evidence (CHE), alters the claim's implied meaning.

4.1 Data Source and Label Schema

The dataset is built upon fact-checking articles from PolitiFact, which include both a concise claim and an accompanying verdict article. Unlike many other fact-checking sources, PolitiFact explicitly considers completeness in its rating criteria: a claim rated *True* must be both accurate and complete, while *Mostly True* and *Half-True* indicate
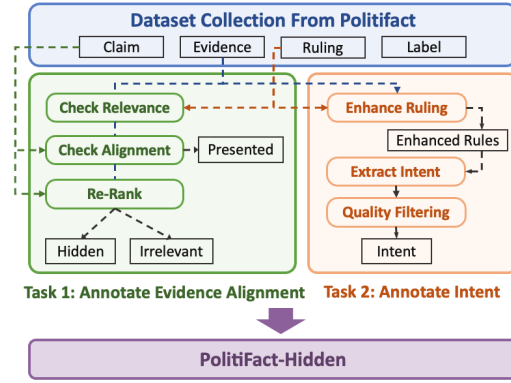

Figure 2: Illustration of the semi-automated annotation pipeline for constructing PolitiFact-HIDDEN, combining `GPT-4o-mini` prompting with human quality control.

factual correctness with missing context (Holan, 2018). In contrast, *Mostly False* reflects the presence of conflicting evidences.

We consolidate PolitiFact's original six-level rating into three coarse-grained labels to align with our half-truth detection task:

Each article is split into evidence paragraphs, which provide factual context, and ruling paragraphs, which justify the final verdict. To prevent label leakage, we separate these segments using structural cues (e.g., "Our Ruling") and exclude ruling content from model input.

To improve generalization and test temporal robustness, we collect an additional 2,000 claims from 2020–2025 to form a temporally disjoint test set. Claims with date overlap are removed from the training pool. The resulting dataset contains 14,994 claims. Detailed statistics are shown in Table 3.

4.2 Annotation Pipeline

Evidence Annotation For each evidence sentence, we determine whether it is already reflected in the claim. This involves:

1. **Relevance Check:** Filter out irrelevant content using LLM-based entailment prompting.

2. **Presentation Check:** Assess whether the content is explicitly or implicitly stated in the claim.

3. **Similarity Refinement:** Use cosine similarity with XLM-RoBERTa embeddings(Nils Reimers, 2019) to refine edge cases and mitigate hallucinations.

Dimension	Requirement	LLM-Positive	Human Confirmed	Agreement
Plausibility	The inferred intent must not contradict the claim.	95	94	98.9%
Implicity	The intent should be implied, not overtly stated.	94	93	98.9%
Sufficiency	The description must be specific and informative.	81	80	98.8%
Readability	The intent must be clearly and fluently expressed.	76	70	92.1%

Table 4: Agreement between LLM and human annotations across intent quality dimensions.

Evidence is labeled as either PE or HE. Manual inspection of 50 samples showed an 88% agreement between LLM predictions and human judgments, validating the alignment process.

Intent Annotation. A key element of half-truth detection is the claim's Intent, i.e., the implied message or judgment it seeks to convey. Intents are extracted in 3 steps:

1. **Ruling Enhancement:** Enhance ruling text by adding supporting evidence for clarity.

2. **Intent Extraction:** Use instruction-tuned prompting to extract the claim's intended conclusion.

3. **Quality Filtering:** Filter extracted intents using four criteria, namely plausibility, implicity, sufficiency and readability.

To validate the quality of LLM-based filtering, we had two human annotators independently assess 100 samples across the same four evaluation dimensions. Agreement between the LLM and both annotators was high (92.1-98.9% across dimensions), suggesting that the LLM-assisted approach reliably captures high-quality intents for downstream reasoning. The full intent evaluation prompts are provided in Appendix A.

5 The TRACER Framework

We propose **TRACER**, a modular framework for detecting half-truths by identifying and evaluating omitted context. TRACER is designed to integrate with existing fact verification (FV) systems by re-assessing claims, particularly those initially labeled as *True*, to determine whether omissions materially alter the claim's intended message.

TRACER operates in three stages: (1) evidence alignment, (2) intent generation, and (3) causal estimation of omitted content. These components support a final re-assessment module that refines the output of base FV models.
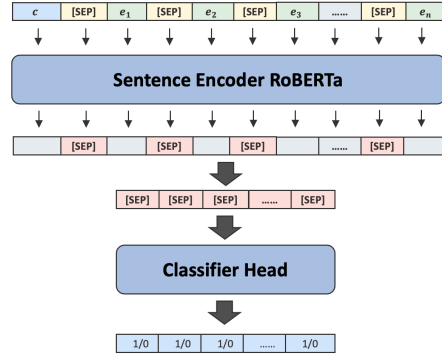

Figure 3: Architecture of the evidence alignment module, which classifies each evidence sentence as presented or hidden relative to the claim.

5.1 Evidence Alignment

The first stage determines whether each evidence sentence $e_i \in E$ is explicitly or implicitly reflected in the claim C. We formulate this as a binary classification task, assigning each e_i to either Presented Evidence (PE) or Hidden Evidence (HE). Only HE is forwarded for further analysis.

As shown in Figure 3, a transformer-based alignment model is adopted. Each (C, e_i) pair is concatenated and encoded using RoBERTa-large (Liu et al., 2019). A classification head predicts whether the evidence content is present in the claim. This alignment step enables TRACER to isolate potentially omitted but relevant information for downstream intent and causal reasoning.

5.2 Intent Generation

Understanding this latent intent is essential for determining whether omitted content is misleading. As described in Section 4.2, we prompt-tune an LLM using input that includes the claim and its associated evidence context to infer intent. This prompt-based formulation encourages the model to extract implicit conclusions without relying on manually predefined templates. The resulting intents serve as semantic anchors for subsequent causality analysis.

5.3 Causality Analysis

While assumptions are derived from HE, not all HE sentences directly affect the plausibility of the intent. Many are tangential or neutral. To distinguish Critical Hidden Evidence (CHE) from neutral omissions, we estimate the causal influence of each HE sentence on the inferred intent.

Inspired by abductive reasoning frameworks (Chen et al., 2022), we generate candidate assumptions A_i that must hold for the intent Z to be valid. These assumptions are derived from evidence through binary question generation and abstraction.

We then evaluate the impact of each A_i using counterfactual prompting: given $do(A_i = \neg A_i)$, does the intent Z still hold? If not, A_i is marked as causally important. For each validated assumption, we retrieve corresponding CHE from the HE pool by selecting sentences that either support or contradict it, based on semantic similarity and an NLI model that verifies logical entailment. This two-step refinement prevents irrelevant or weakly related evidence from being misclassified as CHE.

5.4 Final Re-Assessment Module

To determine the final label (*True*, *Half-True*, or *False*), we incorporate the inferred intent, assumptions, and selected CHE into a re-assessment module (RA). This module re-evaluates the original FV prediction, especially when the claim was initially classified as *True*.

If no CHE is found, the original label is preserved. If CHE alters the plausibility of the intent, the system reclassifies the claim as *Half-True* or *False*, depending on the nature of the conflict. This re-assessment stage is implemented as a prompt-based module. It is designed to be model-agnostic and can be plugged into existing FV pipelines to enhance their ability to detect omission-based manipulation. We provide the full prompt examples used in each component of TRACER in Appendix B.

6 Experiments

We evaluate TRACER by integrating it into existing fact verification (FV) models and measuring its effectiveness in identifying omission-based misinformation. Specifically, we compare TRACER-enhanced models against strong FV systems and conduct ablation studies to assess the impact of individual components. Evaluation metrics include overall Accuracy, macro-F1, and F1 on the *Half-*
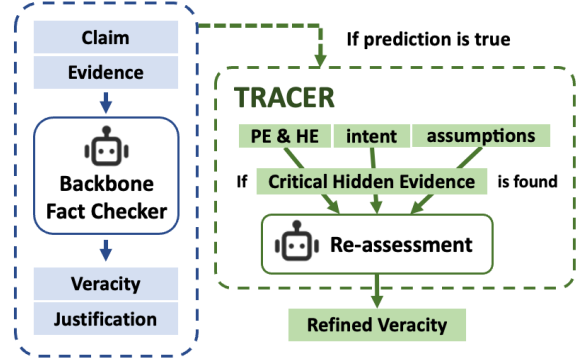

Figure 4: TRACER integrated into a fact verification pipeline as a re-assessment module.

True class (F1(H)), which reflects the system's ability to capture omission-driven misinterpretations.

6.1 Evidence Alignment

We train our evidence alignment model using RoBERTa-large[1], with a context-aware batch sampling strategy. At each training step, sequential evidence segments are grouped into a batch to help the model leverage intra-batch contextual signals. We compare this setup to a baseline where each claim-evidence pair is processed independently (i.e., context-unaware). Both models are trained for 5 epochs with a batch size of 8 and a learning rate of 1e-5.

Method	Accuracy	F1
RoBERTa-large	93.2	90.3
TRACER (context-aware)	**94.0**	**91.6**

Table 5: Evidence alignment performance.

As shown in Table 5, the context-aware training improves F1 by 1.3 and accuracy by 0.8, showing enhanced ability to detect omitted evidence.

6.2 Intent Generation

We fine-tune GPT-4o-mini via the OpenAI API to generate implicit intent statements. Each training input includes the claim and relevant evidence paragraphs. We compare this approach to a 4-shot in-context prompting baseline. The fine-tuned model is trained for 3 epochs with a batch size of 4.

As shown in Table 6, fine-tuning consistently outperforms prompting across all metrics, supporting our decision to use supervised intent extraction in TRACER.

---

[1] https://huggingface.co/FacebookAI/roberta-large

Method	ROUGE-L	BLEU	BERTScore
Few-shot	37.7	6.1	91.2
Fine-tuned	**46.2**	**8.0**	**91.5**

Table 6: Performance of intent generation methods.

Method	Accuracy		F1	
	Dev	Test	Dev	Test
QACheck	48.5	48.8	38.0	38.6
ProgramFC	55.4	56.9	32.9	34.2
CHECKWHY	74.8	65.9	64.2	54.6
Flan-T5	69.7	70.0	50.8	50.4
CoT	76.6	76.3	68.5	64.3
CoT +RA	77.3	78.5	**68.7**	**68.0**
Improvement	*↑ 0.7*	*↑ 2.2*	*↑ 0.2*	*↑ 3.7*
HiSS	76.3	78.3	60.3	59.4
HiSS +RA	**78.1**	**81.9**	64.3	65.7
Improvement	*↑ 1.8*	*↑ 3.6*	*↑ 4.1*	*↑ 6.3*

Table 7: Overall accuracy and macro-F1 on fact verification. RA denotes integration of the TRACER reassessment module.

6.3 Baselines

TRACER requires the fact-checking method to produce justifications for the claim's veracity. This is because TRACER assesses truthfulness by jointly considering the factual accuracy of the claim and the plausibility of its intent, where the former should be supported by explicit reasoning steps. We evaluate TRACER on top of two leading fact verification models that are suitable for integration:

- **Chain-of-Thought (CoT)** (Kojima et al., 2022): a zero-shot prompting baseline, where the model is guided to generate intermediate reasoning steps before producing the final fact-checking verdict.

- **HiSS** (Zhang and Gao, 2023): a state-of-the-art instruction-following verifier that employs structured reasoning by decomposing the claim into multiple verifiable subclaims and evaluating them step by step.

We also report results for the following four baselines:

- **QACheck** (Pan et al., 2023a) and **ProgramFC** (Pan et al., 2023b): agent-based fact-checkers. QACheck decomposes claims into sub-questions and verifies them with evidence. ProgramFC treats verification as a structured program of sub-tasks generated via in-context learning and executed by modular agents.

- **CHECKWHY** (Si et al., 2024) and **Flan-T5** (Chung et al., 2022): prompting-based LLMs. CHECKWHY models causal reasoning through argument structures. Flan-T5 is identified as a strong fact verifier in hallucination evaluations.

To ensure fairness, we evaluate all baselines using `GPT-4o-mini`, except in cases where prior work demonstrates that a different backbone yields stronger performance. For HiSS, we find `GPT-3.5-turbo` consistently outperforms `GPT-4o-mini`.

6.4 Main Results

We present the overall performance of TRACER-integrated models and baselines in Table 7 (Accuracy and macro-F1) and Table 8 (Precision, Recall, and F1 on the Half-True category). The results highlight TRACER's consistent improvements in both general fact verification and the more challenging omission-sensitive cases.

Overall Performance. Table 7 shows that TRACER improves both accuracy and macro-F1 when added to strong reasoning-based backbones. For example, integrating TRACER with HiSS improves test accuracy from 78.3% to 81.9%, and macro-F1 from 59.4 to 65.7. Similarly, CoT benefits from TRACER with a 2.2 point gain in test accuracy and a 3.7-point increase in macro-F1. These gains are observed across both dev and test sets, indicating the robustness of TRACER as a general-purpose re-assessment module.

Half-True Detection. As shown in Table 8, TRACER substantially enhances performance on the Half-True class. When applied to HiSS, TRACER improves F1 by 16.1 points on the test set (from 44.4 to 60.5) and recall by 28.9 points (from 37.9 to 66.8), demonstrating its effectiveness in identifying omission-based manipulation. Similar improvements are seen for CoT, with F1 increasing from 52.8 to 60.2 and recall rising by 15.5 points (from 63.8 to 79.3).

Agent-based baselines such as QACheck and ProgramFC achieve low recall and F1, highlighting their inability to capture hidden context. In contrast, prompting-based methods are more competitive, but still benefit significantly from TRACER's re-assessment. These results validate our hypothesis that omission-aware reasoning, grounded in evidence alignment, intent modeling, and causal

Method	Dev			Test		
	Precision	Recall	F1	Precision	Recall	F1
QACheck	24.0	55.1	33.4	24.3	54.4	33.6
ProgramFC	11.8	2.0	3.5	18.2	6.9	10.0
CHECKWHY	43.2	76.5	55.3	34.3	58.1	43.1
Flan-T5	37.7	33.7	35.6	44.1	27.3	33.7
CoT	44.6	71.8	55.0	45.0	63.8	52.8
CoT +RA	45.5 ↑ *0.9*	83.6 ↑ *11.8*	59.0 ↑ *4.0*	48.5 ↑ *3.5*	79.3 ↑ *15.5*	60.2 ↑ *7.4*
HiSS	34.9	47.6	40.2	53.7	37.9	44.4
HiSS +RA	46.3 ↑ *11.4*	54.4 ↑ *6.8*	50.0 ↑ *9.8*	55.3 ↑ *1.6*	66.8 ↑ *28.9*	60.5 ↑ *16.1*

Table 8: Precision, Recall, and F1 on the Half-True category. TRACER consistently improves detection of omission-based manipulation across all backbones.

Method	True	Half-True	False
CoT	52.9	52.8	87.1
CoT +RA	56.7 ↑ *3.8*	60.2 ↑ *7.4*	87.1 *(–)*
HiSS	44.7	44.4	88.9
HiSS +RA	46.6 ↑ *1.9*	60.5 ↑ *16.1*	90.1 ↑ *1.2*

Table 9: Per-class F1 scores on the test set.

Cfg	Intent	Assump.	Causal.	F1 (H)	F1
①	–	–	–	44.4	59.4
②	✓	–	–	50.9	64.7
③	✓	✓	–	**61.2**	61.7
④	✓	✓	✓	60.5	**65.7**

Table 10: Ablation results for TRACER components.

analysis, substantially improves a model's ability to detect half-truths.

Per-Class Performance. To further examine TRACER's effect on fact verification, we report per-class performance for the top-performing models. As shown in Table 9, TRACER substantially improves the classification of *Half-True* claims while also maintaining or slightly enhancing performance on *True* and *False* claims. This confirms that the observed gains are not achieved at the expense of other classes.

Generalization. To examine the generalization of TRACER, we evaluate it with the open-source LLaMA2-7B model as the base verifier on the top-performing HiSS framework. With TRACER, accuracy improves from 78.2 to 82.3 and Macro-F1 from 59.1 to 65.4. A breakdown of per-class performance and a follow-up analysis of results over different claim lengths is provided in Appendix C.

6.5 Qualitative Analysis

To illustrate how TRACER detects omission-based manipulation, we present representative examples from the POLITIFACT-HIDDEN test set. These cases show how factually accurate claims can still mislead through selective presentation, and how TRACER corrects such misclassifications by identifying Critical Hidden Evidence (CHE) and reasoning about intent.

Example: Misleading Attribution of Rising Costs. *Claim: "Under the Obama economy, utility bills are higher."* This claim was labeled **True** by HiSS, as it aligns with data showing an increase in utility costs during President Obama's term. However, TRACER inferred an intent to attribute blame for rising prices to Obama's economic policies. It then retrieved CHE showing that electricity prices rose even faster under the previous administration and followed a similar pattern across presidencies. This weakened the implied causal attribution and led TRACER to revise the label to Half-True.

Retrieved CHE: *"Rates rose at a significantly faster pace under Bush than they did under Obama." "Trends were not radically different between the Bush and Obama administrations."*

6.6 Ablation Study

We conduct an ablation study to evaluate the contribution of each component within the TRACER framework. Using HiSS as the base verifier, we progressively introduce intent modeling, assumption inference, and causality estimation. Results are shown in Table 10.

Impact of Intent Modeling. Setting ① represents the base HiSS model without any TRACER components. In Setting ②, we introduce intent generation but omit assumption inference and causality estimation. CHE is retrieved directly based on the inferred intent. This setup yields a sub-

stantial improvement in both F1(H) and macro-F1, rising from 44.4 to 50.9 and from 59.4 to 64.7, respectively, demonstrating that intent modeling alone provides meaningful signals for identifying omission-based misdirection.

Assumption Inference. Setting ③ extends the previous configuration by incorporating assumption inference, where the inferred intent is decomposed into finer-grained, testable assumptions. However, causality estimation is still disabled in this setting, meaning that all generated assumptions are treated equally during CHE retrieval. This leads to a further boost in F1(H) to 61.2, validating the utility of breaking down intent into more specific reasoning units. Nonetheless, macro-F1 decreases slightly to 61.7 due to an increase in false positives, indicating that not all assumptions contribute constructively.

Causality Filtering. In Setting ④, our full TRACER framework is applied, with all components enabled, including causality estimation to filter out non-causal or spurious assumptions. While F1(H) drops marginally to 60.5, macro-F1 improves significantly to 65.7. This suggests that causality checking effectively suppresses noisy or irrelevant assumptions, resulting in a more balanced and robust system.

7 Conclusion

This work introduces the task of half-truth detection, addressing claims that are factually correct but misleading due to omitted context. To support this, we introduce POLITIFACT-HIDDEN, a new benchmark with annotated evidence alignment and intent. We propose **TRACER**, a novel framework that detects omission-based misinformation via intent modeling and causal reasoning over hidden content. Integrated with existing fact verification models, TRACER consistently improves performance, especially on half-truths, demonstrating the importance of reasoning about omitted information. This work highlights omission-aware verification as a critical next step for building trustworthy fact-checking systems, and establishes TRACER as a generalizable framework for tackling this underexplored but essential challenge.

Limitations

While TRACER demonstrates strong performance in identifying omission-based misinformation, several limitations remain. First, our evaluation focuses on political discourse, as POLITIFACT-HIDDEN is constructed from the PolitiFact corpus. While TRACER is designed to be model-agnostic and domain-independent, its effectiveness in other domains, such as health or finance, remains to be validated, especially where omission patterns may differ. Second, TRACER assumes that each claim expresses a coherent and inferable intent. However, real-world claims may be vague, ambiguous, or convey multiple overlapping intents, which can introduce noise in downstream reasoning. Future work may explore more robust modeling of claim pragmatics and intent uncertainty to extend TRACER's applicability to broader scenarios.

Acknowledgments

References

Canyu Chen and Kai Shu. 2024. Combating misinformation in the age of llms: Opportunities and challenges. *AI Mag.*, 45(3):354–368.

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. Association for Computational Linguistics.

Brian Jun Rong Chong, Yixuan Tang, and Anthony Kum Hoe Tung. 2025. Mpcg: Multi-round persona-conditioned generation for modeling the evolution of misinformation with llms. In *EMNLP*. Association for Computational Linguistics.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, et al. 2022. Scaling instruction-finetuned language models. *Preprint*, arXiv:2210.11416.

Andrew Estornell, Sanmay Das, and Yevgeniy Vorobeychik. 2020. Deception through half-truths. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(06):10110–10117.

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

Angie Drobnic Holan. 2018. The principles of the truth-o-meter: Politifact's methodology for independent fact-checking. Last updated Jan. 12, 2024.

Israa Jaradat, Haiqi Zhang, and Chengkai Li. 2024. On detecting cherry-picking in news coverage using large language models. *CoRR*, abs/2401.05650.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Jiyoung Lee and Keeheon Lee. 2024. Measuring falseness in news articles based on concealment and overstatement. *Preprint*, arXiv:2408.00156.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Maria D. Molina, S. Shyam Sundar, Thai Le, and Dongwon Lee. 2019. "fake news" is not simply false information: A concept explication and taxonomy of online content. *American Behavioral Scientist*, 65(2):180–212. Original work published 2021.

Iryna Gurevych Nils Reimers. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *https://arxiv.org/abs/1908.10084*.

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.

Raúl Rodríguez-Ferrándiz. 2023. An overview of the fake news phenomenon: From untruth-driven to post-truth-driven approaches. *Media and Communication*, 11(2):15–29.

Michael Sejr Schlichtkrull, Zhijiang Guo, and Andreas Vlachos. 2023. AVeriTeC: A dataset for real-world claim verification with evidence from the web. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Jiasheng Si, Yibo Zhao, Yingjie Zhu, Haiyang Zhu, Wenpeng Lu, and Deyu Zhou. 2024. CHECKWHY: Causal fact verification via argument structure. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15636–15659. Association for Computational Linguistics.

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. *Preprint*, arXiv:2308.07973.

Yixuan Tang, Hwee Tou Ng, and Anthony K. H. Tung. 2021. Do multi-hop question answering systems know how to answer the single-hop sub-questions? In *EACL*, pages 3244–3249. Association for Computational Linguistics.

Yixuan Tang, Yuanyuan Shi, Yiqun Sun, and Anthony Kum Hoe Tung. 2025. Uncovering the bigger picture: Comprehensive event understanding via diverse news retrieval. In *EMNLP*. 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*, pages 809–819.

Bing Wang, Ximing Li, Changchun Li, Bo Fu, Songwen Pei, and Shengsheng Wang. 2024. Why misinformation is created? detecting them by integrating intent features. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, CIKM '24, page 2304–2314, New York, NY, USA. Association for Computing Machinery.

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.

Chang Yang, Peng Zhang, Hui Gao, and Jing Zhang. 2024. Deciphering rumors: A multi-task learning approach with intent-aware hierarchical contrastive learning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 4471–4483. Association for Computational Linguistics.

Zhiwei Yang, Jing Ma, Hechang Chen, Hongzhan Lin, Ziyang Luo, and Yi Chang. 2022. A coarse-to-fine cascaded evidence-distillation neural network for explainable fake news detection. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 2608–2621. International Committee on 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*

A Prompts for Constructing POLITIFACT-HIDDEN

This section presents the prompt templates employed in building the POLITIFACT-HIDDEN dataset.

Prompt: Evidence Relevance Classification

You are tasked to determine the relevance of an evidence to an event.
You will be given a claim, the fact-checking justification of this claim, and an evidence. Is the evidence irrelevant to the event?

Irrelevant: The evidence does not talk about one aspect of the event.
Relevant: The evidence talks about one aspect of the event even if it does not directly address the claim or shares the general topics of the event or simply reference to the original claim.

You do not need to focus on does the evidence support or refute the claim.
Evidence: {evidence}
Justification: {ruling}
Claim: {claim}

Is the evidence relevant to the event?
A. Yes
B. No
Answer only one letter:

Prompt: Evidence Presence Classification

You are tasked to determine whether the evidence is presented in a claim.
You will be given a claim and evidence. Is the evidence presented in the claim?

Presented should satisfy the following conditions:

1. The evidence partly or fully supports the claim. No contradiction is found.

2. The evidence supports the claim without further reasoning, because information is **directly** and **explicitly** stated in the claim.

Evidence: {evidence}
Claim: {claim}

Is the evidence presented in the claim?
A. Yes
B. No
Answer only one letter:

Prompt: Enrich Fact-Checking Ruling with Evidence Given.

You will be provided with the ruling and evidence from a fact-checking article. Your task is to enhance the clarity and depth of the ruling.

Definitions:

- **Ruling:** A concise summary of the fact-checking article that includes the veracity rating of the claim.

- **Evidence:** The supporting details and collected data related to the claim.

Requirements:

- **Identify Ambiguities:** Review the ruling and evidence to pinpoint any unclear or incomplete information in the ruling.

- **Enrich with Evidence:** Refer to the relevant parts of the evidence to expand the ruling. Ensure the enriched ruling explicitly explains how the evidence supports or contradicts the claim and connects directly to its veracity rating.

- **Create a Comprehensive Ruling:** The enhanced ruling should independently present the full context of the fact-checking process and the rationale for the given rating.

Evidence: {evidence}
Ruling: {ruling}

Do not output other thing except your enhanced ruling.

Prompt: Intent Analysis

A claim would convey implicit intents. You are required to determine the intent of a claim based on context in Ruling.
Definition:

- **Claim:** The claim that is checked.

- **Ruling:** Text to determine veracity and explain how the claim would shape people's understanding.

- **Intent:** The understanding of the event that the speaker wants to shape, which is not directly presented in the claim.

(3 Examples are omitted)
Requirements:

1. Intent must be checkable. For example, "people should do something" is not checkable because it does not happen until now.

2. Output intent in <>.

3. Please think step by step. First write your rationale, then the intent.

Claim: {claim}
Ruling: {ruling}

To avoid repetition, we use colors in the prompts to denote different evaluation dimensions, which are assessed independently in practice.

Generated Intent Evaluation (4 dimensions)

You are required to determine whether the intended conclusion is a plausible intent of the claim / conveys the implicit meaning of the claim / readable / sufficient, meaning that it is understandable within the scope of general knowledge.
Please rate using the following scale:

- **0 (not plausible):** The claim contradicts the intended conclusion.

- **1 (plausible):** The claim does not contradict the intended conclusion.

- **0 (not implicit):** The intended conclusion simply rephrases some part of the claim. It does not convey any implicit meaning of the claim.

- **1 (implicit):** The intended conclusion reveals implicit information that is not explicitly stated in the claim.

- **0 (not readable):** The intended conclusion is not readable and is overly complicated.

- **1 (readable):** The intended conclusion is readable and understandable.

- **0 (not sufficient):** The intended conclusion has obvious ambiguous references and is not understandable. For example, it uses unclear terms like "the claim".

- **1 (sufficient):** The intended conclusion is clearly referenced and understandable on its own.

Claim: {claim}
Intended Conclusion: {intent}

Output only one digit.

B Prompt Templates Used in TRACER

This appendix provides the complete prompt templates employed at each stage of the TRACER framework. We include prompts for implicit question generation, assumption inference, causality evaluation, and final re-assessment.

Prompt: Implicit Questions Generation.

A claim can be literally accurate but still misleading in an implicit way.
Your task is to identify the important implicit questions addressed by the evidence.

Steps:

1. Read the evidence below carefully to understand the full context and the topics it covers.

2. Assume the claim is true. What important implicit yes-no questions should be asked to verify the intended conclusion, rather than just the literal accuracy of the claim?

3. Generate 1–3 such implicit questions.

4. Each question should be enclosed in its own angle brackets <>.

5. All implicit questions must be yes-no questions.

(Examples are omitted.)
Claim: {claim}
Intended conclusion: {intent}
Evidence:
{evidence}

A claim could be literally accurate but still misleading because of its intended conclusion.
Your task is to determine what assumptions the intended conclusion is based on, besides the claim.

Definition:

- **Claim:** A statement assumed to be true.

- **Intended conclusion:** The intended conclusion of the claim, which needs checking.

- **Questions:** Some important questions when checking the claim.

- **Assumptions:** The assumptions that the intended conclusion is based on, besides the claim.

Steps:

1. Read the claim, intended conclusion, and questions.

2. Assuming the claim is correct, what assumptions does the question imply should serve as the basis for the intended conclusion?

3. Output a 1–3 sentence rationale, followed by 1–{assumption_max_number} assumptions. Each assumption should be enclosed in angle brackets <> and separated by ||.

Requirements:

1. Ensure that each assumption can independently convey its meaning.
   For example, never use vague references like "the claim," "the evidence," or "the intent"; instead, refer to specific information.

2. Only include assumptions that you are confident in and that serve as a strong basis for the intended conclusion.

(Examples are omitted.)
Claim: {claim}
Intended conclusion: {intention}
Questions:
{questions}

> **Prompt: Causality Analysis**
>
> You are required to do a counterfactual causal inference on a given causal graph.
> **Argument:**
>
> ```
> {
>   "Z": intent,
>   "linked_by": {
>     "X": claim,
>     "Y_1": assumption_1,
>     "Y_2": assumption_2
>   }
> }
> ```
>
> Evaluate $\Delta P(Z \mid \text{do}(\{\text{letter}\} = \neg\{\text{letter}\}))$.
> More specifically, how does the probability of Z change when we set {letter} from {letter} to $\neg$\{letter\}?
> **Options:**
> A. The probability of Z does not change.
> B. The probability of Z increases (Z becomes more likely to be true).
> C. The probability of Z decreases (Z becomes less likely to be true).
> Please answer with **one letter only.**

> **Prompt: Re-Assessment**
>
> A claim may be factually accurate but still misleading due to its implied conclusion. Your task is to refine the veracity assessment of such a claim by considering additional hidden information.
> You are given a previously generated fact-checking **justification**, along with new **evidence** and an **argument** supporting the intended conclusion of the claim.
> Please determine whether the justification has already addressed the hidden information. Then, refine the veracity of the claim accordingly.
> **Input:**
>
> ```
> Evidence: [EVIDENCE]
> Argument: [ARGUMENT]
> Justification: [JUSTIFICATION]
> ```
>
> **Instruction:** Reassess the veracity of the claim based on the above.
> **Choose one of the following options (output only the letter):**
> A. True // B. Half-true // C. False // D. Unverifiable (e.g., the hidden assumption does not support the conclusion, or the information is insufficient)
> **Your answer (one letter only):**

C Generalization with LLaMA2-7B

Results in Table 11 demonstrate that TRACER yields consistent improvements across metrics when applied to the open-source LLaMA2-7B, with particularly notable gains in Half-True classification.

Model	Accuracy	Macro-F1	F1(True)	F1(Half-True)	F1(False)
HiSS	78.3	59.4	44.7	44.4	88.9
HiSS + RA (GPT-3.5-turbo)	81.9	**65.7**	**46.6**	60.5	90.1
HiSS + RA (LLaMA2-7B)	**82.3**	65.4	43.6	**61.3**	**91.2**

Table 11: Generalization of TRACER with different backbones.

We further analyze TRACER's performance across different claim lengths. Using the open-source `LLaMA2-7B` backbone, we partition test claims into four length ranges by word count. Table 12 shows consistent improvements across all ranges, with larger gains observed for longer claims, which likely offer richer context for intent inference and assumption generation.

Model	4–13 (755)	14–23 (893)	24–34 (308)	≥35 (44)
HiSS	80.3	78.5	73.1	75.0
HiSS + RA	**83.7**	**81.8**	**80.5**	**81.8**
Improvement	↑3.4	↑3.3	↑7.5	↑6.8

Table 12: TRACER's performance across different claim lengths (F1 scores) using `LLaMA2-7B`. Numbers in parentheses indicate the number of examples per length range. Longer claims provide richer context, leading to larger improvements.