Foundations of Global Consistency Checking with Noisy LLM Oracles

Anonymous ACL submission

Abstract

Ensuring that collections of natural-language facts are globally consistent is essential for tasks such as fact-checking, summarization, and knowledge base construction. While Large Language Models (LLMs) can assess the consistency of small subsets of facts, their judgments are noisy, and pairwise checks are insufficient to guarantee global coherence. We formalize this problem and show that verifying global consistency requires exponentially many oracle queries in the worst case. To make the task practical, we propose an adaptive divide-and-conquer algorithm that identifies minimal inconsistent subsets (MUSes) of facts and optionally computes minimal repairs through hitting-sets. Our approach has low-degree polynomial query complexity. Experiments with both synthetic and real LLM oracles show that our method efficiently detects and localizes inconsistencies, offering a scalable framework for linguistic consistency verification with LLM-based evaluators.

1 Introduction

Ensuring the *global consistency* of sets of natural-language facts is essential for core NLP applications such as multi-document summarization, fact-checking, and knowledge base construction (Chen et al., 2024; Guo et al., 2022). For example, reports describing the same real-world event may contain overlapping or partially conflicting claims; systems must determine whether all claims can jointly hold, and if not, identify where contradictions arise. Crucially, it is not enough to merely detect that some inconsistency exists. In many downstream pipelines, naïvely discarding all facts whenever a contradiction is detected is unacceptable: a single conflict can cause large numbers of otherwise correct statements to be removed, degrading the quality of summaries, reports, or databases. What is needed instead are *explanations of inconsistency* and *principled ways to repair* fact sets by retaining as many mutually consistent facts as possible while isolating the smallest conflicting groups. This motivates a focus on *Minimal Unsatisfiable Subsets* (MUSes), the smallest sets of claims that cannot jointly be true.

Large Language Models (LLMs) are increasingly used as *judges* for evaluation and verification tasks (Gu et al., 2025; Zhu et al., 2025; Wang et al., 2024a), and they can often assess whether a *small* set of claims appears consistent (Hong et al., 2025; Li et al., 2024). However, pairwise checks do not imply global consistency, and exhaustively querying all subsets is infeasible (Kumar et al., 2023). Moreover, direct "all-at-once" judgments become increasingly unreliable as the number of claims grows, due to longer inputs and denser interactions among statements. The central challenge is therefore: *how can we verify global consistency over many claims while issuing as few noisy LLM-judge calls as possible, and while retaining fine-grained explanations of inconsistency?*

Prior NLP work on factuality, contradiction detection, and fact verification has largely operated at the level of individual claims or pairs—for example, verifying a claim against evidence (Wang and Shu, 2023; Tan et al., 2025), decomposing complex claims (Pan et al., 2023), or retrieving supporting passages (Aly and Vlachos, 2022; de Marneffe et al., 2008). Classic surveys emphasize the importance of factual consistency in NLP systems (Thorne and Vlachos, 2018). However, these approaches do not address *global, multi-statement inconsistency detection*: ensuring that a set of extracted facts is jointly coherent. In real NLP pipelines—such as multi-document retrieval-augmented generation, large-scale information extraction, or report generation—the extracted fact set itself can become contradictory even when each claim is individually supported. From a computational perspective, global consistency is intractable in general (Lee and Leung, 2010), but the presence of small conflict sets or structural regularities makes adaptive, query-efficient approaches viable in practice.

In this paper, we formalize scalable global con-

sistency verification as querying a *noisy subset-consistency oracle*, instantiated by an LLM judge. We show that global consistency cannot be certified from pairwise checks alone and that worst-case query complexity is exponential even under strong assumptions. To make the task practical, we propose an adaptive divide-and-conquer algorithm that localizes Minimal Unsatisfiable Subsets (MUSes) within a set of natural-language claims and, when desired, computes minimal repairs via hitting-set duality. We provide theoretical bounds on query complexity and noise amplification, and empirically demonstrate that the proposed method efficiently detects and explains contradictions using LLMs, substantially improving recall over direct "all-at-once" judging while preserving high precision.

2 Related Work

Local Consistency Verification. Fact and claim consistency verification with LLMs has attracted growing interest, motivated by challenges such as hallucinations and misinformation (Rahman et al., 2025; Singhal et al., 2024). Early methods focused on extracting structured knowledge units such as subject–predicate–object triples from both LLM outputs and reference texts to detect local inaccuracies and support retrieval-augmented verification (Chen et al., 2025; Cao et al., 2025; Lewis et al., 2021). Other approaches decompose responses into atomic claims that are checked independently, which works well for short texts but struggles in long-form or multi-document settings (Hu et al., 2025; Wanner et al., 2024). Reranking methods (Liu et al., 2025) can help mitigate this limitation, but typically require access to model internals, limiting their practical deployment.

Global Consistency and Contradictions. Beyond knowledge-unit extraction, much work has studied factuality and contradiction detection, often casting the problem as a natural-language inference (NLI) task (Thorne et al., 2018; Kryscinski et al., 2020). Benchmarks such as FEVER and VitaminC emphasize identifying local entailment or contradiction, but they do not address whether an entire set of claims can jointly be true (Thorne et al., 2018; Schuster et al., 2021). More recently, LLMs themselves have been used as judges for factuality and coherence (Zheng et al., 2023; Wang et al., 2024b), extending this line of work beyond classifier-based approaches. Concurrent work has studied logical consistency of LLMs on propositional queries over knowledge graphs (Ghosh et al., 2025), e.g. whether $A \wedge B$ is judged consistent with A and B separately, improving performance via fine-tuning.

To our knowledge, our work is the first to formalize *scalable global consistency verification*, prove theoretical query-complexity bounds, and propose adaptive algorithms for isolating minimal inconsistent subsets under noisy LLM oracles. We view this as the first step toward principled, scalable methods for global consistency verification with LLMs.

3 Setting the Stage

Problem Definition. Given a finite fact set $F = \{f_1, \ldots, f_N\}$, we assume that there exists a *ground truth function* $A : 2^F \to \{\mathsf{cons}, \mathsf{incons}\}$, i.e., takes a (sub)set of facts and returns whether it is globally consistent (cons) or not (incons).

Let F be a finite set of facts and $C = \{C_1, \ldots, C_m\}$ a family of scopes with $C_i \subseteq F$. We seek a kept set $F' \subseteq F$ that maximizes coverage while satisfying all per-scope constraints:

$$\max_{F' \subseteq F} |F'| \quad \text{s.t.} \quad A(F' \cap C_i) = \mathsf{cons}, \ \ \forall i \in [m].$$

Define $\widetilde{C}_i \triangleq F' \cap C_i$. Then $\widetilde{C}_i \subseteq C_i$ and $A(\widetilde{C}_i) = \mathsf{cons}$ for all i, and $|\bigcup_{i=1}^m \widetilde{C}_i| = |F'|$. An equivalent formulation of the above objective in terms of minimizing a size of *hitting set* is formulated in Appendix A.1.[1]

Complexity Landscape. In the worst case, solving the above optimization problem is NP-hard, as A can be used to encode any function (such as boolean satisfiability). Note that it is also possible for all fact pairs to be mutually consistent while the full set remains globally inconsistent (see Appendix A.2).

LLM as Noisy Subset-Consistency Oracle. We simulate A using LLMs. Given a finite fact set $F = \{f_1, \ldots, f_N\}$, we model a pretrained LLM as a *noisy subset-consistency oracle (NSCLM)* O as follows. For any subset $S \subseteq F$ we prompt "Are the following claims mutually consistent?" and receive a stochastic response $O(S) \in \{\mathsf{cons}, \mathsf{incons}\}$. Let α, β denote the error rates on the oracle O's performance.

$$\Pr[O(S) = \mathsf{incons} \mid A(S) = \mathsf{cons}] \leq \alpha$$
$$\Pr[O(S) = \mathsf{cons} \mid A(S) = \mathsf{incons}] \leq \beta.$$

When using LLM as O, naively querying $O(F)$ is unreliable in practice as the set size of F increases,

[1]The hitting set problem seeks the smallest subset of elements that intersects every set in a given family.

and as noted above pairwise checks are insufficient to detect inconsistencies.

4 Method

We now formalize our approach to consistency checking under a NSCLM. Our algorithm assumes as input a set of natural-language facts F and a family of *constraints* C with $C_j \subseteq F$ for all $C_j \in C$. Constraints may be given externally (e.g., from a schema or ontology) or constructed automatically from F; here we focus on the given-constraint case for clarity. We start with the definition of Minimal Unsatisfiable Subset (MUS) which forms the basic building block of our approach.

Definition 4.1 (Minimal Unsatisfiable Subset w.r.t. Oracle O)**.** A subset $U \subseteq F$ is an Minimal Unsatisfiable Subset (MUS) if $O(U) = \textsf{incons}$ but $O(U') = \textsf{cons}$ for all proper subsets $U' \subset U$.

Our procedure (Algorithm 1) runs an iterative two-step loop—*MUS extraction* followed by *greedy repair via a hitting set*—repeating until all constraints are consistent , and return the surviving facts F'. The soundness guarantee of the procedure is presented in Appendix A.3.

Assumptions. Our theoretical analysis assumes (i) approximate independence of repeated oracle calls so that majority voting reduces noise, (ii) small conflict size k in practice (typically $k \leq 3$), and (iii) constraint scopes C_i that are either externally defined or automatically constructed from entity or event clusters. These assumptions are used *only* to derive worst-case guarantees and are *not required* by the empirical method: all experiments use a single oracle call per query ($r = 1$) with automatically constructed scopes.

MUS Extraction via Divide-and-Conquer. Our MUS localization procedure builds on the QuickXplain algorithm (Junker, 2004), which given a set of possibly inconsistent constraints, identifies a minimal unsatisfiable subset through a recursive divide-and-conquer strategy. By recursively partitioning the constraint set and reusing intermediate results, QuickXplain achieves logarithmic query growth in subset size. We discuss more details about QuickXplain in Appendix C.

Given an input (O, S, B), where B denotes a background set of facts assumed to be consistent, QuickXplain begins by splitting S into two parts, $S_1 \cup S_2$, and querying the oracle O on $B \cup S_1$. If $O(B \cup S_1) = \textsf{incons}$, it continues by recursing on $(S_1,\ B \cup S_2)$; otherwise, it proceeds with $(S_2,\ B \cup S_1)$. The recursion keeps narrowing the search until $|S| = 1$, ultimately returning a subset-minimal inconsistent set U.

Algorithm 1: QXR

Input: Facts F;
Constraints $C = \{C_1, \ldots, C_m\}$;
Noisy LLM oracle O
Output: Consistent facts F'

$F' \leftarrow F$;
$\mathcal{O}_{\textsf{incons}} \leftarrow \varnothing$;
while $\exists\, C_j \in C : O(C_j) = \textsf{incons}$ **do**
  $\mathcal{U} \leftarrow \varnothing$;
  for $C_j \in C$ ***with*** $O(C_j) = \textsf{incons}$ **do**
    $\mathcal{U} \leftarrow \mathcal{U} \cup \{\mathrm{QX}(O, C_j, \varnothing)\}$
  $H \leftarrow \textsc{GreedyHittingSet}(\mathcal{U})$;
  $F' \leftarrow F' \setminus H$;
  $C \leftarrow \{C_j \setminus H : C_j \in C\}$;
  $\mathcal{O}_{\textsf{incons}} \leftarrow \mathcal{O}_{\textsf{incons}} \cup \mathcal{U}$
return F'

Assume S contains a MUS U of size k. Starting from $\mathrm{QX}(O, S, \varnothing)$, the QuickXplain procedure returns some MUS $U' \subseteq S$ using at most $\mathcal{O}(k \log |S|)$ oracle calls to O.

Greedy Repair via Hitting Set (Minimal Correction Set). Once MUSes are extracted, we identify the inconsistent scopes:

$$T_{\text{incons}} = \big\{j \in [m] : O(C_j) = \textsf{incons}\big\}. \qquad (1)$$

For each $j \in T_{\text{incons}}$, we obtain a MUS $U_j \subseteq C_j$, and form the family of conflicts $\mathcal{U} = \{U_j : j \in T_{\text{incons}}\}$. We then compute a *repair set* $H \subseteq F$ that intersects every MUS:

$$\forall U \in \mathcal{U}, \quad H \cap U \neq \varnothing. \qquad (2)$$

Such a minimal hitting set H corresponds exactly to a *Minimal Correction Set (MCS)*—the smallest subset of facts whose removal restores global consistency. We remove H to obtain the consistent subset $F' = F \setminus H$. Intuitively, the hitting set selects the fewest facts that "break" all discovered inconsistencies. For example, if $\mathcal{U} = \{\{a, b, c\}, \{a, d, e\}\}$, then any H intersecting both conflicts is valid, and the minimal hitting set $H = \{a\}$ yields the maximal consistent subset $F' = \{b, c, d, e\}$. In practice we use a greedy solver that iteratively selects the fact covering the largest number of uncovered MUSes, achieving the optimal logarithmic approximation ratio for this NP-hard problem but can be approximated efficiently (see Appendix B.

Let N be number of facts in F, $m = |C|$ be the number of constraints, k the maximum size of

	VitaminC						FEVER					
	Direct (baseline)			QXR (ours)			Direct (baseline)			QXR (ours)		
Model	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
Claude 3.7 (Anthropic, 2025a)	**0.979**	0.854	0.909	0.956	**0.975**	**0.965**	**0.992**	0.805	0.873	0.983	**0.977**	**0.980**
Claude 4 (Anthropic, 2025b)	**0.956**	0.877	0.913	0.938	**0.983**	**0.960**	**0.995**	0.833	0.891	0.981	**0.977**	**0.978**
DeepSeek-R1 (DeepSeek-AI et al., 2025)	**0.980**	0.730	0.827	0.973	**0.990**	**0.981**	**0.989**	0.821	0.875	0.988	**0.980**	**0.983**
GPT-OSS-120B (OpenAI, 2025)	**0.984**	0.926	0.953	0.956	**0.995**	**0.975**	0.976	0.975	0.976	**0.992**	**0.980**	**0.985**
Mistral Large (2407) (AI, 2023)	0.955	0.603	0.724	**0.968**	**0.978**	**0.972**	**0.970**	0.780	0.848	0.964	**0.990**	**0.976**

Table 1: **Evaluation of consistent fact sets F' on two datasets (VitaminC (Schuster et al., 2021) / FEVER (Thorne et al., 2018)).** Precision (P), recall (R), and F1 are computed with respect to gold consistent subsets. QXR yields cleaner and more complete F' than direct all-at-once LLM judging.

any MUS discovered by Algorithm 1, and I be the number of outer rounds of Algorithm 1 until termination.

Theorem 4.2 (Query Complexity of Algorithm 1)**.** *Algorithm 1 makes at most $I \cdot m \cdot (k \log N)$ oracle calls to LLM O.*

5 Experiments

5.1 Experiment Setting.

Datasets We use VitaminC (Schuster et al., 2021) and FEVER (Thorne et al., 2018) to global consistency by grouping 30 factual statements per example. In VitaminC, each cluster contains 24–28 compatible claims and 2–6 injected contradictions from `REFUTES` edits, each forming a size-2 ground-truth MUS $\{c^+, c^-\}$. In FEVER, we combine 0–8 `REFUTES` claims (with evidence) with 14–18 `SUPPORTS` claims; each refuting claim and its evidence define a ground-truth MUS. Removing either the refuting claim or its evidence yields consistent variants, producing dense contradiction structures that require multi-claim reasoning.

Direct LLM Consistency. We first consider a *direct-LLM baseline* that queries the model once over the entire fact set F, prompting it to return the largest subset $F' \subseteq F$ that is jointly consistent. This corresponds to treating the LLM as an unstructured oracle $O(F)$ that attempts to approximate the ground-truth function $A(F)$ in a single step.

Evaluation. Given an initial fact set F, the model produces a repaired subset F' after removing claims identified as inconsistent.We evaluate the resulting F' against the **gold consistent subset** F_{gold}, defined as the maximal subset of F that contains no injected contradictions (i.e., all ground truth satisfiable claims). Precision, recall, and F1 are computed on the surviving facts:

$$P = \frac{|F' \cap F_{\text{gold}}|}{|F'|}, \qquad R = \frac{|F' \cap F_{\text{gold}}|}{|F_{\text{gold}}|}.$$

This directly measures how accurately the model preserves all and only the consistent information, matching the formal objective in Theorem A.1. All experiments use a single oracle call per query ($r = 1$) and the simplest scope setting $C = F$, without majority voting or independence assumptions; such assumptions are used only in the theoretical analysis.

5.2 Results and Analysis.

Table 1 shows consistent gains from MUS-based reasoning. Across all models, QXR yields cleaner consistent subsets F' with substantially higher F1 by avoiding the over-removal seen in direct all-at-once prompting, which often "panic-prunes" large clusters once any conflict is detected. By adaptively isolating minimal conflicts and repairing only what is necessary, QXR preserves nearly all valid information while restoring global consistency. We observe the same trend on a synthetic dataset (Direct: P=0.515, R=0.993, F1=0.644; QXR: P=0.664, R=0.878, F1=0.724), indicating more targeted inconsistency identification (see Appendix E).

All experiments use a zero-shot LLM judge. Evaluating the direct baseline with Chain-of-Thought, decomposition, few-shot, and self-consistency prompting yields the same failure mode: high precision but low recall, showing robustness to prompt design (see Appendix F).

6 Conclusion

We introduced the task of *global fact consistency verification* under noisy LLM oracles, established limits on pairwise sufficiency and query complexity, and proposed an adaptive algorithm that localizes minimal inconsistent subsets and repairs them via hitting-set. On VitaminC clusters, the method improves recall and F1 while preserving high precision, showing that structured querying can turn LLMs into scalable consistency checkers. Future work will extend to larger knowledge graphs and integrate with retrieval and summarization pipelines.

Limitations

Our theoretical analysis assumes repeated oracle queries can reduce noise under approximate independence. In practice, however, LLM errors may be systematic rather than random, and repeated queries to the same model do not necessarily improve reliability. For this reason, our empirical evaluation does not rely on majority voting or repeated queries: all experiments use a single LLM call per query ($r = 1$) with the simplest scope setting ($C = F$). We treat the independence assumption solely as a modeling abstraction for deriving worst-case guarantees.

A related limitation is that noise reduction in real systems may require querying *diverse* LLMs rather than repeatedly querying the same model. Exploring ensemble or cross-model consistency checking is a promising direction for future work, but is beyond the scope of this paper.

Finally, our experiments focus on moderate-sized fact sets constructed from existing benchmarks. While these settings already expose substantial failures of direct LLM judging, larger and more heterogeneous fact collections—such as those arising in long-context RAG or large-scale knowledge extraction—may introduce additional challenges. Designing benchmarks that better capture such regimes remains an open problem.

References

Mistral AI. 2023. Mixtral of experts: release of mixtral 8×7b. https://mistral.ai/news/mixtral-of-experts.

Rami Aly and Andreas Vlachos. 2022. Natural logic-guided autoregressive multi-hop document retrieval for fact verification. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6123–6135, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Anthropic. 2025a. Claude 3.7 sonnet and claude code. https://www.anthropic.com/news/claude-3-7-sonnet. Accessed: 2025-10-06.

Anthropic. 2025b. Introducing claude 4. https://www.anthropic.com/news/claude-4.

Han Cao, Lingwei Wei, Wei Zhou, and Songlin Hu. 2025. Enhancing multi-hop fact verification with structured knowledge-augmented large language models. *Preprint*, arXiv:2503.08495.

Ting-Chih Chen, Chia-Wei Tang, and Chris Thomas. 2024. MetaSumPerceiver: Multimodal multi-document evidence summarization for fact-checking. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8742–8757, Bangkok, Thailand. Association for Computational Linguistics.

Yingjian Chen, Haoran Liu, Yinhong Liu, Jinxiang Xie, Rui Yang, Han Yuan, Yanran Fu, Peng Yuan Zhou, Qingyu Chen, James Caverlee, and Irene Li. 2025. Graphcheck: Breaking long-term text barriers with extracted knowledge graph-powered fact-checking. *Preprint*, arXiv:2502.16514.

Marie-Catherine de Marneffe, Anna N. Rafferty, and Christopher D. Manning. 2008. Finding contradictions in text. In *Proceedings of ACL-08: HLT*, pages 1039–1047, Columbus, Ohio. Association for Computational Linguistics.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 181 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Bishwamittra Ghosh, Sarah Hasan, Naheed Anjum Arafat, and Arijit Khan. 2025. Logical consistency of large language models in fact-checking. In *The Thirteenth International Conference on Learning Representations*.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, Saizhuo Wang, Kun Zhang, Yuanzhuo Wang, Wen Gao, Lionel Ni, and Jian Guo. 2025. A survey on llm-as-a-judge. *Preprint*, arXiv:2411.15594.

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

Zhaochen Hong, Haofei Yu, and Jiaxuan You. 2025. ConsistencyChecker: Tree-based evaluation of LLM generalization capabilities. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 33039–33075, Vienna, Austria. Association for Computational Linguistics.

Qisheng Hu, Quanyu Long, and Wenya Wang. 2025. Decomposition dilemmas: Does claim decomposition boost or burden fact-checking performance? In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6313–6336, Albuquerque, New Mexico. Association for Computational Linguistics.

Ulrich Junker. 2004. Quickxplain: Preferred explanations and relaxations for over-constrained problems. pages 167–172.

Wojciech Kryscinski, Bryan McCann, Caiming Xiong, and Richard Socher. 2020. Evaluating the factual consistency of abstractive text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9332–9346, Online. Association for Computational Linguistics.

Sujit Kumar, Rohan Jaiswal, Mohit Ram Sharma, and Sanasam Ranbir Singh. 2023. Multiset dual summarization for incongruent news article detection. In *Proceedings of the 20th International Conference on Natural Language Processing (ICON)*, pages 779–790, Goa University, Goa, India. NLP Association of India (NLPAI).

Jimmy Lee and K. Leung. 2010. A stronger consistency for soft global constraints in weighted constraint satisfaction. *Proceedings of the AAAI Conference on Artificial Intelligence*, 24:121–127.

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. 2021. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Preprint*, arXiv:2005.11401.

Haitao Li, Qian Dong, Junjie Chen, Huixue Su, Yujia Zhou, Qingyao Ai, Ziyi Ye, and Yiqun Liu. 2024. Llms-as-judges: A comprehensive survey on llm-based evaluation methods. *Preprint*, arXiv:2412.05579.

Tianyu Liu, Jirui Qi, Paul He, Arianna Bisazza, Mrinmaya Sachan, and Ryan Cotterell. 2025. Pointwise mutual information as a performance gauge for retrieval-augmented generation. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1628–1647, Albuquerque, New Mexico. Association for Computational Linguistics.

OpenAI. 2025. Introducing gpt-oss. `https://openai.com/index/introducing-gpt-oss/`.

Liangming Pan, Xiaobao Wu, Xinyuan Lu, Anh Tuan Luu, William Yang Wang, Min-Yen Kan, and Preslav Nakov. 2023. 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.

Subhey Sadi Rahman, Md. Adnanul Islam, Md. Mahbub Alam, Musarrat Zeba, Md. Abdur Rahman, Sadia Sultana Chowa, Mohaimenul Azam Khan Raiaan, and Sami Azam. 2025. Hallucination to truth: A review of fact-checking and factuality evaluation in large language models. *Preprint*, arXiv:2508.03860.

Tal Schuster, Adam Fisch, and Regina Barzilay. 2021. Get your vitamin C! robust fact verification with contrastive evidence. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 624–643, Online. Association for Computational Linguistics.

Aryan Singhal, Thomas Law, Coby Kassner, Ayushman Gupta, Evan Duan, Aviral Damle, and Ryan Luo Li. 2024. Multilingual fact-checking using LLMs. In *Proceedings of the Third Workshop on NLP for Positive Impact*, pages 13–31, Miami, Florida, USA. Association for Computational Linguistics.

Xin Tan, Bowei Zou, and Ai Ti Aw. 2025. Improving explainable fact-checking with claim-evidence correlations. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 1600–1612, Abu Dhabi, UAE. Association for Computational Linguistics.

James Thorne and Andreas Vlachos. 2018. Automated fact checking: Task formulations, methods and future directions. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3346–3359, Santa Fe, New Mexico, USA. 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.

Vijay V Vazirani. 2001. *Approximation algorithms*, volume 1. Springer.

Binjie Wang, Steffi Chern, Ethan Chern, and Pengfei Liu. 2024a. Halu-j: Critique-based hallucination judge. *Preprint*, arXiv:2407.12943.

Haoran Wang and Kai Shu. 2023. Explainable claim verification via knowledge-grounded reasoning with large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6288–6304, Singapore. Association for Computational Linguistics.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, Wayne Xin Zhao, Zhewei Wei, and Jirong Wen. 2024b. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6).

Miriam Wanner, Seth Ebner, Zhengping Jiang, Mark Dredze, and Benjamin Van Durme. 2024. A closer look at claim decomposition. In *Proceedings of the 13th Joint Conference on Lexical and Computational Semantics (*SEM 2024)*, pages 153–175, Mexico City, Mexico. Association for Computational Linguistics.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Lianghui Zhu, Xinggang Wang, and Xinlong Wang. 2025. Judgelm: Fine-tuned large language models are scalable judges. *Preprint*, arXiv:2310.17631.

A Theorems and Proofs

Let $F = \{f_1, \ldots, f_N\}$ be a finite fact set, $C = \{C_1, \ldots, C_m\}$ with $C_i \subseteq F$ be a collection of constraint scopes, and a perfect oracle $O : 2^F \to \{\mathsf{cons}, \mathsf{incons}\}$ that returns whether a subset of facts is jointly consistent. Throughout the proofs, we will use two standard properties.

- **Monotonicity**: If $U \subseteq S \subseteq F$ and $O(U) = \mathsf{incons}$ then $O(S) = \mathsf{incons}$.
- **Existence of MUS**: If $O(S) = \mathsf{incons}$ and S is finite then S contains a MUS $U \subseteq S$.

These follow from finiteness and the definition of minimality.

A.1 Objective Equivalence and Reduction to Hitting Set

Theorem A.1. *Objective Equivalence Maximizing coverage*

$$\max_{F' \subseteq F} |F'| \quad s.t. \quad A(F' \cap C_i) = \mathsf{cons}. \quad \forall i \in [m] \tag{3}$$

is equivalent to minimizing deletions

$$\min_{R \subseteq F} |R| \quad s.t. \quad A((F \setminus R) \cap C_i) = \mathsf{cons}, \quad \forall i \in [m] \tag{4}$$

Proof. Define a bijection between solutions $R = F \setminus F'$ and $F' = F \setminus R$. Then

$$|F'| = |F| - |R|. \tag{5}$$

Hence, maximizing $|F'|$ is equivalent to minimizing $|R|$. Under this bijection, the feasibility constraints are identical. □

Let $\mathcal{A}_{\mathsf{incons}}$ be the family of all MUSes possible in C. Formally, based on Definition 4.1, we define

$$\mathcal{A}_{\mathsf{incons}} = \{U \in C_j : j \in [m],\ U \text{ is a MUS w.r.t. } A\}.$$

Theorem A.2. *A set $R \subseteq F$ is feasible iff it is a hitting set for $\mathcal{A}_{\mathsf{incons}}$.*

Proof. In one direction, assume $R \subseteq F$ is feasible for minimum deletion. Suppose for contradiction, there exists $U \in \mathcal{A}_{\mathsf{incons}}$ with $R \cap U = \varnothing$. By definition of $\mathcal{A}_{\mathsf{incons}}$, there is some i with $U \subseteq C_i$, and thus $U \subseteq (F \setminus R) \cap C_i$. Since $A(U) = \mathsf{incons}$ by monotonicity $(F \setminus R) \cap C_i$ would be inconsistent, contradiction to Equation (4). Hence, $R \cap U \neq \varnothing$ for all $U \in \mathcal{A}_{\mathsf{incons}}$; i.e., R is a hitting set.
For the other direction, assume R is a hitting set for $\mathcal{A}_{\mathsf{incons}}$, i.e. $R \cap U \neq \varnothing$ for all $U \in \mathcal{A}_{\mathsf{incons}}$. Suppose for contradiction that R is not feasible; then there exists some i with $A((F \setminus R) \cap C_i) = \mathsf{incons}$. By the existence-of-MUS property, $(F \setminus R) \cap C_i$ contains a MUS $U \subseteq (F \setminus R) \cap C_i$. Then $U \in \mathcal{A}_{\mathsf{incons}}$ but $U \cap R = \varnothing$, contradicting that R hits all MUSes. Hence $A((F \setminus R) \cap C_i) = \mathsf{cons}$ for all i, i.e. R is feasible. □

A.2 Pairwise Checks Are Insufficient for Global Consistency

Theorem A.3 (Pairwise Insufficiency)**.** *Even with access to A, there exist F of size $N \geq 3$ such that all pairs are consistent, but F is inconsistent globally.*

Proof. Given A we now want to show pairwise consistency does not imply global consistency. Let the universe be boolean assignments to variables $A, B, C \in \{0, 1\}$. Consider the three facts $f_1 : A \oplus B = 1$, $f_2 : B \oplus C = 1$, $f_3 : C \oplus A = 1$. Then, any two pairs can be satisfied, for example f_1, f_2 are satisfied with $A = 1, B = 0, C = 1$. However, this is jointly unsatisfiable, from f_1 and f_2: $A = \neg B$ and $C = \neg B$ hence $A = C$, then $C \oplus A = 0$ which contradicts f_3. Hence, $f_1, \wedge f_2, \wedge f_3$ is inconsistent. We can think of this as a graph colouring problem, for example the constraints $A \neq B, B \neq C, C \neq A$ requires a 2-colouring of a 3-cycle, which is impossible. □

A.3 Soundness of Algorithm 1

Theorem A.4 (Soundness under Perfect Oracle)**.** *Assume that the LLM oracle O has zero error ($\alpha = 0, \beta = 0$), then for every $i \in [m]$, the retained subset $F' \triangleq F \setminus R$ satifies*

$$A(F' \cap C_i) = \mathsf{cons}. \tag{6}$$

Proof. If O is perfect, then it should match the groun-truth A on all inputs. Suppose for contradiction that $O(F' \cap C_i) = \mathsf{incons}$ for some $i \in [m]$. By MUS existence, there is a MUS $U \subseteq F' \cap C_i$. Then, $U \subseteq C_i$, so $U \in \mathcal{A}_{\mathsf{incons}}$. We know $U \subseteq F \setminus R$ so we know $U \cap R = \varnothing$ contradicting R hits every $U \in \mathcal{A}_{\mathsf{incons}}$. Therefore, $O(F' \cap C_i) = \mathsf{cons}$ for all i. □

In practical scenarios, the MUSes $\mathcal{O}_{\mathsf{incons}}$ extracted by Algorithm 1 may contain errors, as the procedure depends on a noisy oracle O. However, F' generated by in Algorithm 1 is still sound with respect to the MUSes found.

Theorem A.5 (Practical Soundness w.r.t. Extracted Conflicts)**.** *Let $\mathcal{O}_{\mathsf{incons}}$ be the set of MUSes actually extracted by the Algorithm 1 under O. Let $R \subseteq F$ satisfy $R \cap U \neq \varnothing$ for all $U \in \widehat{\mathcal{U}}$. The retained set $F \setminus R$ is consistent with respect to the extracted conflicts $\mathcal{O}_{\mathsf{incons}}$.*

Proof. For every $U \in \mathcal{O}_{\mathsf{incons}}$, $R \cap U \neq \varnothing$ implies $U \not\subseteq F \setminus R$. Thus none of the discovered MUSes in $\mathcal{O}_{\mathsf{incons}}$ remain after removal. □

A.4 Error Reduction under Repetition

Let the true perfect oracle answer for a query be $Y \in \{\mathsf{cons}, \mathsf{incons}\}$. A noisy oracle O is repeatedly queried r times on the same set S. Each repetition returns $\widehat{Y}_t \in \{\mathsf{cons}, \mathsf{incons}\}$, $t = 1, \ldots, r$. For each repetition and conditioning on the true label Y, we get

$$\Pr(\widehat{Y}_t \neq Y | Y) \leq \varepsilon \quad \varepsilon \triangleq \max\{\alpha, \beta\} < \frac{1}{2} \quad (7)$$

We assume independence across repetitions, e.g., $\{\widehat{Y}_t\}_{t=1}^r$ are independent conditioned on Y. We also assume r is odd.

Theorem A.6 (Error Reduction Under Repetition)**.** *Let $\mathcal{O}$ be an (α, β)–noisy subset-consistency oracle with $\max\{\alpha, \beta\} < \frac{1}{2}$. If each query is evaluated r times independently and aggregated by majority vote, the effective error rate per aggregated call is at most $\exp(-2r\gamma^2)$ where $\gamma = \frac{1}{2} - \max\{\alpha, \beta\}$.*

Proof. The majority vote is wrong iff at least half of the repetitions are wrong, using a conservative threshold we get

$$\{\widehat{Y}^{maj} \neq Y\} \subseteq \{S_r \geq r/2\}$$

where $X_t = \mathbb{I}\{\widehat{Y}_t \neq Y\}$ and $S_r = \sum_{t=1}^r X_t$. So we can apply Hoeffding's inequality to S_r (sum of independent Bernoulli variables with means $\leq \varepsilon \triangleq \max\{\alpha, \beta\} < \frac{1}{2}$. For any $a > 0$, we have

$$\Pr(S_r - \mathbb{E}[S_r] \geq a) \leq \exp(-\frac{2a^2}{r})$$

here $\mathbb{E}[S_r] = \sum_{t=1}^r \mathbb{E}[X_t] \leq r\varepsilon$. Set $a = r(\frac{1}{2} - \varepsilon) = r\gamma$. Then,

$$\Pr(S_r \geq r/2) \leq \Pr(S_r - \mathbb{E}[S_r] \geq r(\frac{1}{2} - \varepsilon)) \leq \exp(-2r\gamma^2).$$

Algorithm 2: QuickXplain (QX) for MUS Extraction

Input: Oracle O;
Candidate set S;
Background B (assumed consistent)
Output: Subset-minimal inconsistent set $\Delta \subseteq S$ (or $\varnothing$)

if $S = \varnothing$ **then**
  return $\varnothing$
if $O(B \cup S) = \mathsf{cons}$ **then**
  return $\varnothing$
if $|S| = 1$ **then**
  return S
Split S into two (nearly) equal parts S_1, S_2;
$\Delta_1 \leftarrow \mathrm{QX}(O, S_1, B \cup S_2)$;
$\Delta_2 \leftarrow \mathrm{QX}(O, S_2, B \cup \Delta_1)$;
return $\Delta_1 \cup \Delta_2$

Combining together we get per-call bound

$$\Pr(\widehat{Y}^{maj} \neq Y) \leq \exp(-2r\gamma^2).$$

□

B Hitting Set Approximation

Lemma B.1 (Greedy Hitting Set Approximation (Vazirani, 2001))**.** *Let $\mathcal{U}$ be a family of m MUSes (conflict sets) over facts F. The greedy hitting set algorithm returns a hitting set H satisfying $|H| \leq (1 + \ln m)\,|H^\star|$, where $H^\star$ is an optimal (minimum-cardinality) hitting set of $\mathcal{U}$.*

C QuickXplain

QuickXplain (QX) is a classic divide-an-conquer method for localizing a minimal unsatisfiable subset (MUS) from an inconsistent set S under a consistency oracle $O : 2^F \to \{\mathsf{cons}, \mathsf{incons}\}$ (Junker, 2004). The algorithm adaptively queries subsets to find a subset-minimal inconsistent core with logarithmic depth in $|S|$. We provide a pseudocode in Algorithm 2. QX narrows the inconsistent subset by recursively testing halves of S. If $B \cup S_1$ is inconsistent, the conflict lies in S_1; otherwise it lies in S_2. Recursion stops once singleton conflicts are reached, yielding a subset-minimal inconsistent set. For a perfect oracle, QX requires $\mathcal{O}(k \log |S|)$ queries to locate a conflict of size k.

D Pairwise/NLI Baselines are not comparable at scale

Methods that decide consistency by evaluating all sentence pairs require $\binom{N}{2}$ oracle calls. For typi-

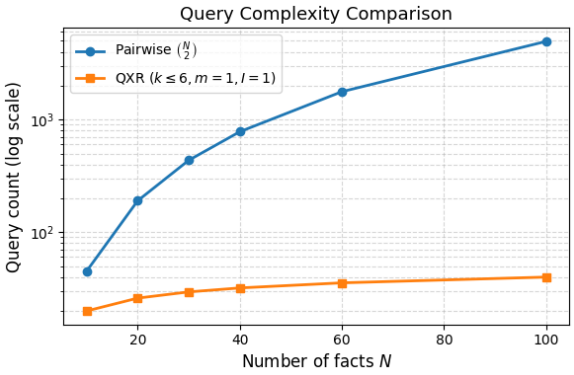

Figure 1: Scaling of query counts with number of facts N. Pairwise checking is quadratic, while QXR scales polylogarithmically

cal sizes of facts e.g., $N \in [30, 100]$, that implies $435 - 4950$ calls *per instance*. Under noisy LLM oracles, each decsion further requires r repetitions (majority vote). Our QXR algorithm performs at most $I \cdot m \cdot (k \log N)$ number of queries where k is the MUS maximum size (empirically small), m is the constraint scope (if no constraints or in general $m = 1$) and I is the number of outer rounds.In our experiments $k \le 6$, $m = 1$, we show a plot how complexity scales in Figure 1 We also illustrate common baselines and why they do not scale in Table 2.

E Synthetic Dataset

Entities and predicates. We sample entity names for `PERSON`, `ORG`, `LOC`,`EVENT`, and `ANIMAL`. Facts are rendered from a small predicate bank with negation support, including unary categories (`IsTiger`, `IsDog`, `IsActor`, `IsPolitician`, `IsAnimal`), binary relations (`WorksFor`, `LocatedIn`), and temporal precedence (`Before`). Numeric paraphrases (`AtLeastCases`, `AtMostCases`) provide clutter.

Planted MUS patterns. We inject one or more MUSes per instance:

- size-2 contradiction $(A, \neg A)$,
- size-3 temporal cycle,
- size-3 "exactly one" parity conflict consisting of a rule sentence (*Exactly one of X, Y, Z holds*) plus two of the three facts.

Distractors. We add on-topic true facts and a small fraction of off-topic false facts (e.g., *does not work for*, *is not located in*) to create realistic clutter.

Gold Annotations. For each instance we store the gold MUS family $\mathcal{U}_{\text{gold}}$as lists of sentence IDs. The gold consistent subset F_{gold} is defined as the maximal consistent subset obtained by removing a minimum hitting set over $\mathcal{U}_{\text{gold}}$ (greedy approximation).

F Prompting Analysis

All main experiments in the paper use a zero-shot LLM-judge setting. To assess sensitivity to prompt design, we additionally evaluated the direct-judge baseline on VitaminC using several advanced prompting strategies. Across all prompt styles, we observe the same qualitative trend: high precision but substantially lower recall due to over-removal of consistent facts. These experiments were run on a subset of the dataset; for reference, we also restate the original zero-shot baseline and QXR results from the full evaluation.

Claude-3.7 (Sonnet)

Prompting Style	P	R	F1
Zero-shot	0.986	0.681	0.797
Chain-of-Thought	0.973	0.684	0.792
Decomposition	0.987	0.626	0.755
Few-shot	0.975	0.658	0.775
Self-consistency	0.986	0.547	0.694
Original zero-shot baseline	0.979	0.854	0.909
Original QXR	0.956	0.975	0.965

Claude-4 (Sonnet)

Prompting Style	P	R	F1
Zero-shot	0.989	0.928	0.954
Chain-of-Thought	0.987	0.908	0.942
Decomposition	0.677	0.608	0.638
Few-shot	0.992	0.891	0.932
Self-consistency	0.992	0.768	0.860
Original zero-shot baseline	0.956	0.877	0.913
Original QXR	0.938	0.983	0.960

DeepSeek-R1

Prompting Style	P	R	F1
Zero-shot	0.993	0.860	0.910
Chain-of-Thought	0.996	0.850	0.907
Decomposition	0.996	0.882	0.930
Few-shot	0.882	0.761	0.813
Self-consistency	0.987	0.718	0.817
Original zero-shot baseline	0.980	0.730	0.827
Original QXR	0.973	0.990	0.981

GPT-OSS-120B

Prompting Style	P	R	F1
Zero-shot	0.994	0.876	0.920
Chain-of-Thought	0.995	0.851	0.903
Decomposition	0.954	0.816	0.865
Few-shot	0.997	0.896	0.932
Self-consistency	0.996	0.475	0.624
Original zero-shot baseline	0.984	0.926	0.953
Original QXR	0.956	0.995	0.975

Method	Description	Complexity
Pairwise NLI Graph (FEVER-style)	Run NLI on every pair of claims, removing any node involved in a contradiction edge.	$O(N^2)$
Transitive Closure / Entailment Graph	Build a full entailment–contradiction graph and perform reasoning (e.g., SAT solving or closure).	$> O(N^2)$
LLM-as-NLI	Prompt an LLM with two sentences (e.g., "Does A contradict B?").	$O(N^2)$
Multi-premise NLI	Treat the entire set of claims as premises and ask if they jointly entail a hypothesis.	$O(1)$
Clustered Pairwise	For each claim, compare only to its top-K nearest neighbors (embedding-based).	$O(NK)$

Table 2: Common NLI-style baselines for consistency checking and their computational complexity. Pairwise and entailment-graph methods grow quadratically with the number of claims, making them infeasible for large clusters. Multi-premise NLI corresponds to our baseline.

Mixtral-8×7B

Prompting Style	P	R	F1
Zero-shot	0.947	0.568	0.701
Chain-of-Thought	0.997	0.570	0.700
Decomposition	0.991	0.535	0.673
Few-shot	0.238	0.142	0.175
Self-consistency	0.990	0.522	0.674
Original zero-shot baseline	0.955	0.603	0.724
Original QXR	0.968	0.978	0.972

G Prompts

This appendix lists the exact prompts used in our experiments. All prompts were intentionally kept simple and symmetric across methods to isolate algorithmic effects rather than prompt engineering. For all baselines that output a subset of facts, models are required to return the *full text* of each retained fact (not indices) as a Python list enclosed in an <answer> tag.

Subset-consistency oracle (QXR). Given a queried subset of facts (optionally with a background set B), we ask the LLM to judge whether all statements can be true simultaneously:

```
Factual statements. Some may contradict.

{bg}Statements:
{facts_block}

Respond ONLY:
- CONSISTENT
- INCONSISTENT

Answer:
```

Direct baseline (zero-shot). The LLM is asked to return a mutually consistent subset of facts:

```
Given the following factual statements,
some may contradict.
Return a Python list of facts that
are mutually consistent, meaning all
returned facts can be true at the same
time.

Facts:
{facts_block}

CRITICAL: Return ONLY a Python
list using the FULL TEXT of each fact.
<answer>["fact1", "fact2",
...]</answer>
```

Direct baseline (Chain-of-Thought prompting). We encourage structured reasoning while constraining the output format:

```
Given the following facts, some
may contradict. Find all mutually
consistent facts.

Facts:
{facts_block}

Think step-by-step:
1. Identify pairs of facts that
contradict each other.
2. For each contradiction, decide which
fact to keep.
3. Return the consistent subset.

CRITICAL: In the <answer> tag, return
the FULL TEXT of each fact, NOT numbers.
Example: <answer>["The sky is blue",
"Grass is green"]</answer>
```

Direct baseline (Decomposition prompting). The task is decomposed into detection and resolution:

```
Task: Select a mutually consistent
subset of the following facts.

Facts:
{facts_block}

Step 1 - Identify contradicting
facts.
Step 2 - Decide which facts to keep.
Step 3 - Output the consistent subset.

CRITICAL: Return ONLY a Python
list with the FULL TEXT of each fact.
<answer>["full fact text 1", "full fact
text 2", ...]</answer>
```

Direct baseline (Few-shot prompting). We provide two illustrative examples followed by the target instance:

```
Given facts that may contradict, return
the subset that is mutually consistent.

Example 1:
- The sky is blue
- The sky is red
- Grass is green
<answer>["The sky is blue", "Grass is
green"]</answer>

Example 2:
- Paris is in France
- Paris has 2 million people
- Paris has 10 million people
<answer>["Paris is in France", "Paris
has 2 million people"]</answer>

Now solve:
Facts:
{facts_block}

<answer>
```

Direct baseline (Self-consistency prompting). We sample multiple outputs using the same prompt and aggregate by majority vote over selected facts:

```
Given the following facts, some may
contradict.
Return ONLY a Python list of facts that
are mutually consistent.

Facts:
{facts_block}

<answer>["fact1",              "fact2",
...]</answer>
```