Global Consistency Repair for Natural-Language Claim Sets

Paul He[1]* **Elke Kirschbaum**[2] **Shiva Prasad Kasiviswanathan**[2]
[1]NTU Singapore [2]Amazon
paul005@ntu.edu.sg elkeki@amazon.com
kasivisw@gmail.com

Abstract

NLP systems increasingly aggregate claims from retrieved passages, documents, and generated reasoning steps. Yet current factuality methods mostly verify claims individually or compare pairs of claims, leaving open the question of whether the entire aggregated claim set can be retained coherently. This misses a fundamental failure mode: claims may be locally plausible and pairwise compatible while still being jointly inconsistent. We formalize global consistency repair: given an inconsistent natural-language claim set, preserve the largest coherent subset while localizing the compact sources of inconsistency. We introduce QXR, an adaptive diagnosis-and-repair framework that treats an LLM as a noisy set-level consistency judge, localizes minimal conflicting subsets, and repairs the claim set by deleting a small hitting set of claims. Across LLM judges and benchmarks derived from VitaminC, FEVER, and conflicting-evidence RAG, QXR preserves substantially more valid claims than direct all-at-once judging while maintaining high precision. These results suggest that factuality in claim aggregation systems should be treated as a set-level repair problem, not only as local verification.[1]

1 Introduction

Natural-language systems increasingly assemble *sets of claims* from multiple documents, retrieved passages, or extracted evidence (Chen et al., 2024; Schlichtkrull et al., 2023; Aly and Vlachos, 2022). Applications such as multi-document summarization, fact checking, retrieval-augmented generation (RAG) (Lewis et al., 2020), and knowledge-base construction must combine information originating from many sources into a coherent final output. Yet individually plausible claims need not be *jointly coherent*: a RAG system may retrieve passages that are each relevant to a query while still supporting incompatible answers (Wang et al., 2025). The downstream challenge is therefore not only to detect that a contradiction exists, but to identify *where* it arises while preserving as much valid information as possible. A key difficulty in these settings is that conflicts are often sparse: only a small subset of claims may be mutually incompatible, while most retrieved or generated information remains useful.

To make this concrete, consider a retrieval-augmented system generating a compliance or incident report from dozens of retrieved passages. Individual statements may each be locally supported while still being jointly incompatible: different documents may disagree on dates, attribute conflicting causes to the same event, or provide mutually inconsistent entity facts. In such settings, discarding all retrieved information is undesirable, while retaining every locally plausible statement can yield incoherent outputs. The practical objective is therefore to preserve the largest subset of claims that remains jointly coherent while isolating only the conflicting information. This paper asks a different question from standard factuality evaluation. Instead of asking whether a single claim is supported, or whether two claims contradict, we ask: given an inconsistent set of natural-language claims, which claims should an NLP system keep? This turns factuality for claim aggregation into a preservation-oriented repair problem.

We study this problem as *global consistency repair*: given a collection of claims, retain the largest subset that remains jointly coherent. Unlike contradiction detection, this objective requires both identifying conflicts and repairing them while preserving unaffected information. To operationalize this goal, we draw inspiration from inconsistency diagnosis: Minimal Unsatisfiable Subsets (MUSes), the smallest groups of claims that cannot jointly be true, provide localized explanations of inconsistency and naturally support targeted repair.

Most NLP approaches to factuality and contradiction operate *locally*: they verify individual claims against evidence, decompose complex

*Work begun while at Amazon Research, Tübingen.
[1]Code available at https://github.com/Hepaul7/QXR

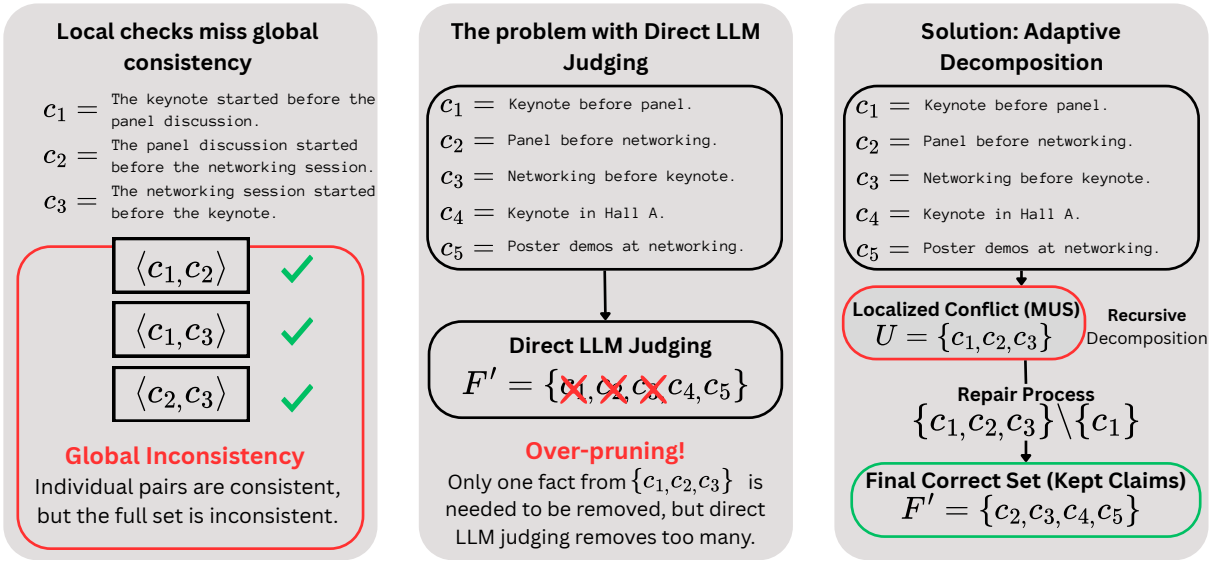

Figure 1: Why global consistency requires structured reasoning. Left: pairwise checks can all succeed even when the full claim set is globally inconsistent. Middle: direct all-at-once LLM judging may over-prune, removing more claims than necessary to restore consistency. Right: adaptive decomposition localizes a minimal unsatisfiable subset and repairs the set while preserving more valid claims.

claims into smaller units, or detect contradictions between pairs of statements (Wang and Shu, 2023; Tan et al., 2025; Pan et al., 2023; de Marneffe et al., 2008). These methods are useful, but they do not solve the set-level problem studied here. A collection of claims may pass local verification and remain pairwise compatible while still being jointly inconsistent. Consequently, local factuality pipelines cannot in general certify that an aggregated evidence set or final system output is globally coherent. This gap becomes increasingly important in long-context RAG, multi-document information extraction, and report generation, where outputs depend on reasoning over an entire claim set rather than isolated statements.

Large language models (LLMs) provide a practical interface for set-level consistency judging: modern models can often assess whether a small subset of claims is mutually compatible (Li et al., 2024), though LLMs are known to be logically inconsistent on complex queries (Ghosh et al., 2025). Furthermore, exhaustive subset checking is infeasible, pairwise checks cannot certify global coherence, and direct "all-at-once" prompting becomes brittle as claim sets grow. We therefore formalize *global consistency repair* as a distinct NLP capability and introduce QXR (Query-eXplain-and-Repair), an adaptive diagnosis-and-repair framework that uses LLM consistency judgments to localize compact conflicting subsets and remove only the claims needed to restore coherence. We provide theoretical motivation for structured decomposition and show empirically, across factuality benchmarks and realistic RAG settings, that QXR preserves substantially more valid claims than direct all-at-once judging while maintaining high precision. Figure 1 illustrates the core failure mode and repair strategy.

2 Related Work

Local Factuality and Claim Verification. Fact verification and factual consistency checking with LLMs have attracted growing attention, motivated by challenges such as hallucinations, misinformation, and unreliable generation (Rahman et al., 2025; Singhal et al., 2024; Guo et al., 2022). One line of work focused on extracting structured knowledge units, such as subject–predicate–object triples, from model outputs and reference texts to identify local inaccuracies and support retrieval-augmented verification (Chen et al., 2025; Cao et al., 2025). Other methods decompose responses into atomic claims that are verified independently (Hu et al., 2025; Wanner et al., 2024), which works well for short responses but becomes increasingly brittle in long-form or multi-document settings where interactions among claims matter. Reranking and retrieval-based methods (Liu et al., 2025) can partially mitigate these issues, but often depend on model internals or scoring signals that are unavailable in black-box LLM settings.

Set-Level Consistency in NLP Systems. Beyond local factuality, substantial work has studied contradiction detection and coherence, often

through natural-language inference (NLI) formulations (Thorne et al., 2018; Kryscinski et al., 2020). Benchmarks such as FEVER and VitaminC focus on detecting whether individual claims are supported or contradicted by evidence (Thorne et al., 2018; Schuster et al., 2021), but do not address whether an *entire set* of claims can jointly remain coherent. More recently, LLMs have been used directly as judges for factuality and coherence (Zheng et al., 2023), and earlier work has also studied logical consistency of LLMs on propositional or knowledge-graph reasoning tasks (Ghosh et al., 2025). Prior work that repairs set-level inconsistency either materializes constraints before solving – e.g., hand-authored rules (Kassner et al., 2021), pairwise-NLI (Mitchell et al., 2022), generated explanations (Jung et al., 2022) – or learns a supervised set-level scorer (Song et al., 2025). Our work takes neither route: it treats consistency judging as a black-box oracle and localizes conflicts through adaptive subset queries, which is what allows it to surface conflicts that no pairwise constraint set can represent, without task-specific training.

Positioning of Our Work. To our knowledge, this is the first work to formalize *global consistency repair* as a constrained optimization over natural-language claim sets: maximizing retained claims subject to scope-wise consistency, characterized through hitting-set duality and query complexity. This preservation-oriented objective is central to RAG, multi-document summarization, and evidence aggregation, where outputs depend on the joint coherence of many interacting claims. Methodologically, QXR combines inconsistency diagnosis with LLM-based consistency judgments, localizing compact conflict sources and removing only the claims needed to restore coherence. Although the components we used are established tools for diagnosing over-constrained systems, what changes here is the setting: the constraints are natural-language claims with no symbolic encoding, and the consistency test is a noisy black-box judge, hence oracle calls are the dominant cost rather than solver time. We now formalize this objective as the preservation-oriented counterpart to local contradiction detection.

3 Problem: Global Consistency Repair

Problem Definition. Let F be a finite set of **facts** and $C = \{C_1, \ldots, C_m\}$ be a **family of scopes** with $C_i \subseteq F$, where each scope groups facts whose joint consistency matters. For example, a scope may correspond to a retrieved evidence set in RAG, a cluster of claims in multi-document summarization, or an entity-specific subset of facts in knowledge aggregation.

We assume an underlying **consistency function** $A : 2^F \to \{\mathsf{cons}, \mathsf{incons}\}$ which determines whether a subset of facts is jointly coherent. Our objective is to **retain** a subset $F' \subseteq F$ that maximizes coverage (i.e., preserves as much information as possible) while remaining consistent within every scope:

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

That is, the objective is explicitly *preservation-oriented*: rather than rejecting an entire inconsistent claim set, we seek the *largest subset of claims that remains jointly coherent* under all relevant scopes. This distinction is important in NLP systems, where downstream outputs often depend on broad evidence coverage. Overly aggressive filtering may discard many correct statements, while insufficient filtering leaves contradictions unresolved. Equivalently, the problem can be viewed as removing the fewest claims necessary to restore coherence. The corresponding minimum-deletion formulation is given in Appendix A.1. We also show in Appendix A.2 that the minimum-deletion view is equivalent to a hitting-set problem over the family of all MUSes.

Definitions. We briefly introduce terminology used throughout the paper. A subset $U \subseteq F$ is a **minimal unsatisfiable subset (MUS)** if $A(U) = \mathsf{incons}$, but every proper subset $U' \subset U$ satisfies $A(U') = \mathsf{cons}$. Intuitively, a MUS is a subset-minimal localized group of claims responsible for a consistency failure. These subsets provide compact explanations of *where* inconsistency arises.

A **repair** is a deletion set $R \subseteq F$ such that the retained facts $F \setminus R$ remain consistent within all relevant scopes. If $\mathcal{U}$ is a family of discovered MUSes, then a set $H \subseteq F$ is a **hitting set** if $H \cap U \neq \varnothing$ for every $U \in \mathcal{U}$. In our setting, removing a hitting set guarantees that at least one claim is removed from each identified conflict, thereby restoring consistency. Thus, MUSes provide localized explanations of inconsistency, while hitting sets provide the corresponding repair mechanism.

LLM-based Consistency Judging. In practice, the consistency function A is not directly observable, and constructing domain-specific symbolic verifiers for open-ended natural-language claims is often impractical. We therefore approximate consistency judgments using a pretrained LLM.

For any subset of claims $S \subseteq F$, we prompt the model to determine whether the claims in S are mutually coherent, yielding a consistency judgment $O(S) \in \{\mathsf{cons}, \mathsf{incons}\}$.

LLMs provide a practical interface for set-level consistency checking because they can often reliably assess whether *small* groups of claims appear mutually compatible. At the same time, these judgments are imperfect: responses may be stochastic, sensitive to prompt formatting, or degrade as the number of interacting claims grows. We therefore view O as a noisy approximation to the underlying notion of global coherence. For theoretical analysis, we treat A as an idealized monotone consistency function and consider assumptions such as perfect judging only to analyze the structure of the problem and provide correctness guarantees. The empirical method itself does not rely on these assumptions.

3.1 Problem Hardness

Global consistency repair presents two challenges. Exact repair is computationally hard, and the relevant inconsistencies may depend on interactions that cannot be detected locally. We establish both points before motivating QXR 's adaptive decomposition.

Worst-Case Hardness. In the worst case, solving the maximum-preservation objective is NP-hard. The ground-truth consistency function A can encode arbitrary satisfiability structure, so exact repair can require solving a general constraint-satisfaction problem. Equivalently, let $\mathcal{U}$ denote the family of all minimal unsatisfiable subsets of F. A deletion $R \subseteq F$ is feasible exactly when it intersects every $U \in \mathcal{U}$. Thus, exact repair can be viewed as finding the smallest set of claims whose removal breaks all conflicts, that is, a minimum hitting set over the discovered inconsistencies.

Pairwise Consistency is Not Enough. Pairwise consistency does not imply global consistency. For any $N \geq 3$, there exists a fact set F of size $\binom{N}{2}$ such that every pair of facts in F is jointly consistent, yet F itself is globally inconsistent. One way to view the construction is through a graph-coloring abstraction: pairwise incompatibility constraints can all be satisfiable in isolation, while the full set requires more global structure than the available assignments permit. This formalizes a core limitation of current fact-checking and factuality pipelines: methods that verify claims individually, or only compare claim pairs, cannot in general certify that a full set of claims is jointly coherent. Full proof details are given in Appendix A.3.

Worst-Case Global Interactions. The obstruction can also be genuinely global. Consider Boolean variables $z_1, \ldots, z_{N+1}$ and fixed bits $x_1, \ldots, x_N \in \{0, 1\}$, with claims of the form $z_{i+1} = z_i \oplus x_i$ for each $i \in [N]$, together with the boundary claim $z_{N+1} = z_1$. Composing the constraints shows that the set is satisfiable iff $\bigoplus_{i=1}^{N} x_i = 0$. Thus, flipping any single x_i changes the global consistency label. In the worst case, consistency can therefore behave like parity: every claim may be pivotal, and the inconsistency may not be attributable to any small local check. This illustrates why direct all-at-once judging can require aggregating long-range interactions across the entire claim set. This observation is consistent with analyses in Hahn and Rofin (2024) showing that parity-like functions are difficult for transformers to model robustly, although those results should not be interpreted as a direct lower bound for decoder-only LLM judges.

Naïve Decomposition Fails; Adaptive Helps. The analysis so far suggests that global consistency cannot, in general, be recovered by simply decomposing a claim set into smaller parts and aggregating the results independently. Suppose an inconsistent claim set $S \subseteq F$ is partitioned as $S = S_1 \cup S_2$. Then at least one of the following must hold: (i) S_1 is inconsistent, in which case some MUS is fully contained within S_1; (ii) S_2 is inconsistent, so some MUS is fully contained within S_2; or (iii) both S_1 and S_2 are individually consistent, yet their union $S_1 \cup S_2$ is inconsistent, implying that the relevant MUS crosses the partition and depends on claims from both subsets. This distinguishes our setting from decomposable query problems, where answers on disjoint subsets can be combined to obtain the answer on their union, enabling generic reductions to efficient algorithms. In global consistency repair, by contrast, this is not possible due to case (iii). The one-MUS case is the simplest counterexample showing that naïve decomposition can fail. With multiple MUSes, each MUS may lie within one partition or cross the partition, and a single crossing MUS is enough to invalidate independent decomposition.

Naïve divide-and-conquer strategies or local verification pipelines may therefore miss contradictions that emerge only across groups of claims. At the same time, the existence of localized conflicts suggests that *structured* decomposition can still be useful: the system should recursively localize compact sources of inconsistency while preserv-

ing globally coherent information. This motivates the adaptive diagnosis-and-repair framework introduced in the next section.

4 Conflict Diagnosis and Repair

We now describe QXR, our framework for global consistency repair using LLM-based consistency judgments. As motivated by the decomposition trichotomy above, QXR uses each split not to aggregate independent answers, but to decide whether to recurse into one side or preserve cross-partition context during localization. Formally, QXR takes as input a fact set F, a family of scopes $C = \{C_1, \ldots, C_m\}$ with $C_i \subseteq F$, and an LLM-based consistency judge O. It returns a repaired subset $F' \subseteq F$ and, optionally, the localized conflicts $\mathcal{U}_{\text{found}}$ discovered during repair. QXR alternates between localizing MUSes and removing a small hitting set until all relevant scopes are judged consistent. By localizing these conflict regions, QXR avoids the over-pruning behavior often observed in direct all-at-once judging. Formal guarantees for soundness, approximation, and query complexity are given in Appendix A.

4.1 Diagnosis-and-Repair Overview

Algorithm 1 gives the overall QXR procedure. Starting from $F' \leftarrow F$, QXR repeatedly identifies the scopes currently judged inconsistent,

$$\mathcal{T}_{\mathsf{incons}} \triangleq \{\, C_j \in C : O(C_j) = \mathsf{incons} \,\},$$

localizes one MUS $U_j \subseteq C_j$ for each such scope, and collects the resulting conflict family

$$\mathcal{U} \triangleq \{\, U_j : C_j \in \mathcal{T}_{\mathsf{incons}} \,\}.$$

These discovered conflicts are also accumulated into $\mathcal{U}_{\text{found}}$, which may optionally be returned as a record of surfaced inconsistencies.

QXR then computes a repair set H that intersects every conflict: $\forall U \in \mathcal{U}, H \cap U \neq \varnothing$. It removes H from the retained fact set and restricts all scopes accordingly. The process repeats until no scope remains inconsistent. Thus, QXR alternates between diagnosis, which exposes compact conflict witnesses, and repair, which removes only facts needed to break those conflicts.

4.2 Conflict Localization

QXR assumes access to a MUS-localization routine LOCALIZEMUS(O, S, B), where $S \subseteq F$ is the candidate set being minimized and B is a background context assumed to satisfy $O(B) = \text{cons}$. The goal is to return a subset $U \subseteq S$ such that $B \cup U$ is inconsistent, but removing any element of U restores consistency relative to B. Thus, U is a subset-minimal explanation of inconsistency conditioned on the current background.

The background context is essential for handling cross-partition conflicts. If an inconsistent candidate $S = S_1 \cup S_2$ satisfies $O(B \cup S_1) = \text{incons}$, then localization can recurse into S_1 while keeping B fixed; the case for S_2 is analogous. The harder case is when $O(B \cup S_1) = O(B \cup S_2) = \mathsf{cons}$ but $O(B \cup S_1 \cup S_2) = \mathsf{incons}$. Here, no conflict is contained entirely within either side relative to B, so the inconsistency requires claims from both S_1 and S_2. Hence, the localizer minimizes one side while treating the other as background, then minimizes the remaining side relative to the reduced context. This conditional use of background distinguishes MUS localization from naïve divide-and-conquer.

We use QuickXplain (Junker, 2004) as the conflict-localization routine. The algorithm uses divide-and-conquer to identify subset-minimal inconsistencies. QuickXplain serves as a replaceable localization primitive; QXR is the surrounding NLP repair framework that combines noisy LLM consistency judgments, conflict localization, and preservation-oriented hitting-set repair. Under a perfect consistency judge, QuickXplain localizes a conflict of size k within a candidate set S using $\mathcal{O}(k \log |S|)$ consistency queries. Its recursive conditioning naturally aligns with the cross-partition conflict structure described above. We defer pseudocode to Appendix B.

4.3 Minimal Repair via Hitting Sets

Given localized conflicts $\mathcal{U}$, any valid repair must remove at least one claim from each conflict. QXR therefore seeks a hitting set $H \subseteq F$ such that $H \cap U \neq \varnothing$ for every $U \in \mathcal{U}$, and returns the repaired claim set $F' = F \setminus H$. This separates diagnosis from repair: localization identifies compact explanations of inconsistency, while the hitting set selects a small deletion set that breaks all discovered conflicts and preserves the remaining claims.

We instantiate this step with a greedy hitting-set solver, which is efficient and has the standard logarithmic approximation guarantee (Lemma A.1). When few conflicts are discovered, the same optimization can instead be solved exactly by enumeration or integer programming. Subsequent consistency checks detect conflicts not yet localized.

4.4 Query Complexity

Because LLM consistency judgments are the dominant computational cost, an important question is

Algorithm 1: QXR: Global Consistency Diagnosis and Repair

Input: Facts F; scopes $C = \{C_1, \ldots, C_m\}$; noisy oracle O
Output: Repaired fact set F'; optionally discovered conflicts $\mathcal{U}_{\text{found}}$

$F' \leftarrow F$;
$\mathcal{U}_{\text{found}} \leftarrow \varnothing$;
while $\exists\, C_j \in C$ *s.t.* $O(C_j) = \textsf{incons}$ **do**
  $\mathcal{U} \leftarrow \varnothing$;
  $\mathcal{T}_{\textsf{incons}} \leftarrow \{\, C_j \in C : O(C_j) = \textsf{incons} \,\}$;
  foreach $C_j \in \mathcal{T}_{\textsf{incons}}$ **do**
    $U_j \leftarrow \textsc{LocalizeMUS}(O, C_j, \varnothing)$;
    $\mathcal{U} \leftarrow \mathcal{U} \cup \{U_j\}$;
  $\mathcal{U}_{\text{found}} \leftarrow \mathcal{U}_{\text{found}} \cup \mathcal{U}$;
  $H \leftarrow \textsc{GreedyHittingSet}(\mathcal{U})$;
  $F' \leftarrow F' \setminus H$;
  $C \leftarrow \{\, C_j \setminus H : C_j \in C \,\}$;
return F' *and optionally* $\mathcal{U}_{\text{found}}$

how many oracle calls QXR requires in practice. The following result characterizes the complexity of the diagnosis-and-repair loop.

Theorem 4.1 (Query Complexity of Algorithm 1)**.** *Let $N = |F|$ be the number of claims, $m = |C|$ the number of scopes, k the maximum size of any MUS localized by* QXR*, and I the number of outer iterations until termination. Assuming* LOCALIZEMUS *is instantiated with QuickXplain, Algorithm 1 makes at most $\mathcal{O}(I \cdot m \cdot (1 + k \log N))$ consistency queries. For $k \geq 1$, this simplifies to $\mathcal{O}(I \cdot m \cdot k \log N)$.*

A proof is given in Appendix A.9. The dependence on k is important: query complexity scales with the size of the localized conflicts rather than the full claim set.

Although the number of scopes m can be exponential in the worst case, many NLP settings involve only a small number of natural scopes. An important observation is that QXR does not enumerate the full family of MUSes, which can itself be exponential in size. Instead, QXR adaptively localizes conflicts only in currently inconsistent scopes, repairs the claim set, and continues diagnosis only if inconsistency remains as judged by the oracle. Therefore, the number of oracle calls depends on the conflicts encountered during repair and their localized size k, rather than on enumerating all possible conflicts. When conflicts are sparse and localized ($k \ll N$), this can further reduce the cost of conflict localization.

	VitaminC	FEVER
MUSes / inst.	4.05 [2, 6]	4.77 [0, 28]
Min. HS size	4.05 [2, 6]	4.01 [0, 8]
Facts / inst.	28.11 [24, 32]	33.73 [20, 62]

Table 1: **Conflict-structure statistics.** Values report mean [min, max] across instances.

To illustrate the above point, consider the fully global setting $\mathcal{C} = \{F\}$. If a scope contains another scope, consistency of the larger scope already implies consistency of the smaller one, making the nested scope redundant. We formalize this as *scope subsumption* in Appendix C. Hence, when the entire claim set F is the scope of interest, we can take $m = 1$, reducing the bound to $\mathcal{O}(I \cdot k \log N)$. In this case, the explicit dependence on the claim-set size N is only logarithmic. This does not imply logarithmic scaling overall, as k and I can still grow with N in the worst case. Although pairwise comparison—requiring $\binom{N}{2} \in \mathcal{O}(N^2)$ checks—provably cannot solve the set-level problem, it is useful to illustrate how quickly the number of oracle calls can scale with N: 435 calls at $N = 30$ and 4,950 at $N = 100$.

Additional structure can also make exact global consistency repair more tractable when $m > 1$. Fact-disjoint scope components can be repaired independently. More generally, consider the conflict graph induced by the family of scope-contained MUSes, where two facts are adjacent if they occur together in some MUS. If this conflict graph has bounded treewidth t, exact repair can be solved in $2^{t+1} \cdot \text{poly}(|F| + |\mathcal{A}_{\text{incons}}|)$ time, given $\mathcal{A}_{\text{incons}}$ and a tree decomposition of width t. Intuitively, low conflict-graph treewidth means that inconsistencies involve localized interactions among facts, rather than many facts being densely interconnected through overlapping conflicts. Thus, the exponential dependence is on the treewidth of the conflict graph rather than the total number of claims (Appendix C).

5 Experiments

5.1 Experimental Setup

Datasets. We construct global-consistency instances from VitaminC (Schuster et al., 2021) and FEVER (Thorne et al., 2018) by grouping approximately 30 factual claims per example. Because benchmark resources for *set-level* consistency verification remain limited, we adapt existing fact-checking datasets into controlled testbeds for global consistency repair. This is a natural set-

	VitaminC						FEVER					
	Direct (baseline)			QXR (ours)			Direct (baseline)			QXR (ours)		
Model	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
Sonnet 3.7	**0.979**	0.854	0.912	0.956	**0.975**	**0.965**	**0.992**	0.805	0.889	0.983	**0.977**	**0.980**
Sonnet 4	**0.956**	0.877	0.915	0.938	**0.983**	**0.960**	**0.995**	0.833	0.907	0.981	**0.977**	**0.978**
Sonnet 4.5	**0.997**	0.617	0.762	0.975	**0.957**	**0.966**	**0.990**	0.901	0.944	0.978	**0.947**	**0.962**
Opus 4.6	0.654	0.574	0.611	**0.985**	**0.968**	**0.976**	**0.995**	**0.984**	**0.989**	0.971	0.958	0.964
DeepSeek-R1	**0.980**	0.730	0.837	0.973	**0.990**	**0.981**	**0.989**	0.821	0.897	0.988	**0.980**	**0.983**
GPT-OSS-120B	**0.984**	0.926	0.954	0.956	**0.995**	**0.975**	0.976	0.975	0.976	**0.992**	**0.980**	**0.985**
Mistral Large 2	0.955	0.603	0.739	**0.968**	**0.978**	**0.972**	**0.970**	0.780	0.865	0.964	**0.990**	**0.976**

Table 2: **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.

ting: both datasets contain semantically related claims with supporting and contradicting evidence, allowing us to study scenarios where individual claims may appear locally plausible while the overall claim set remains jointly inconsistent. The resulting setup enables controlled evaluation of both consistency restoration and *preservation of valid claims*.

In VitaminC, each cluster contains 24–32 claims in total, including 2–6 injected contradictions derived from `REFUTES` edits. Each contradiction forms a size-2 ground-truth MUS $\{c^+, c^-\}$. In FEVER, we combine 0–8 `REFUTES` claims (with evidence) with 14–18 `SUPPORTS` claims. Ground-truth MUSes pair refuting claims with each evidence set they contradict; removing either the refuting claim or its evidence restores consistency, producing dense multi-claim interactions that require set-level reasoning. Table 1 summarizes the resulting conflict structure. VitaminC instances have a relatively uniform number of disjoint conflicts, whereas FEVER is more heterogeneous and can contain multiple MUSes sharing claims, resulting in fewer required deletions than MUSes in some instances.

Models. We evaluate both direct all-at-once judging and QXR using a diverse set of LLM consistency judges: Claude Sonnet 3.7, Claude Sonnet 4, Claude Sonnet 4.5, Claude Opus 4.6, DeepSeek-R1, GPT-OSS-120B, and Mistral Large (2407) (Anthropic, 2025a,b,d, 2026; Guo et al., 2025; OpenAI, 2025; Mistral AI team, 2024). This selection spans both state-of-the-art closed models and strong open-weight models, thus allowing us to test whether the global-consistency gap persists across diverse model families. All experiments use Amazon Bedrock with temperature 0.0 and maximum output length 8192 tokens. We use deterministic decoding to reduce variance across runs and apply the same zero-shot consistency prompt across methods unless otherwise stated.

Direct Baseline. We compare against a *direct all-at-once baseline* that prompts the LLM once over the entire claim set F and asks it to return the largest jointly coherent subset $F' \subseteq F$. This baseline reflects the most natural practical alternative to QXR: rather than localizing conflicts explicitly, the model attempts to resolve global inconsistencies through a single holistic judgment. Our central question is whether structured diagnosis improves the *preservation of valid claims* relative to this direct approach. We discuss why exhaustive pairwise or NLI-style baselines are not computationally comparable at scale in Appendix D.

QXR Setting. Unless otherwise stated, all QXR experiments use a single oracle call per query ($r = 1$) and the scope setting $C = \{F\}$. This is the fully global setting: the entire claim set must be judged jointly consistent, and under monotonicity all nested subscopes are redundant (Appendix C.2). We do not use majority voting or repeated querying in the empirical evaluation; assumptions about repeated-query independence are introduced only in the theoretical analysis. This isolates the contribution of structured conflict localization and repair without relying on additional sampling or self-consistency mechanisms. Prompt templates for both the subset-consistency oracle and the direct baseline are provided in Appendix E.

Evaluation. Given an initial claim set F, the model returns a repaired subset F' after removing claims identified as inconsistent. We evaluate F' against the **gold coherent subset** F_{gold}, defined as

the maximal subset of F containing no injected contradictions (i.e., all ground-truth satisfiable claims). Precision, recall, and F1 are computed over the retained claims:

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

We identify the set of valid minimum repairs and evaluate a prediction against the best-matching valid gold repair, therefore equivalent optimal repairs are not penalized. QXR 's greedy hitting-set solver may also admit tied choices, which we resolve deterministically. This evaluation directly measures the preservation objective: high recall reflects retaining valid claims, while high precision ensures inconsistent claims are removed, matching the formal objective explained in Theorem A.1.

5.2 Results and Analysis

Direct Judging Tends to Over-prune. Table 2 shows the results of experiments for VitaminC and FEVER. The dominant failure mode of direct all-at-once judging is not low precision but low recall: once an inconsistency is detected, the model often removes substantially more claims than are necessary to restore coherence. We refer to this behavior as *over-pruning*. This failure mode is not eliminated by stronger prompting: in Appendix F, direct judging under chain-of-thought, decomposition, few-shot, and self-consistency prompts still tends to preserve high precision but lower recall, indicating that the issue is the lack of explicit conflict localization rather than prompt sensitivity. Because QXR first localizes compact conflicts and then removes only a hitting set of inconsistent claims, it directly targets this failure mode. As a result, QXR typically preserves substantially more valid claims while maintaining competitive precision.

The gains are especially pronounced on VitaminC. For example, Claude Sonnet 4.5 improves from $R = 0.617$ to $R = 0.957$, while Claude Opus 4.6 improves from $R = 0.574$ to $R = 0.968$. One possible reason is VitaminC's contrastive construction, as it is built from Wikipedia revisions where supporting and refuting evidence differ by small factual edits (Schuster et al., 2021). Therefore, contradictions are often subtler and depend on fine-grained factual details, which may make direct all-at-once repair harder even on frontier models. On FEVER, conflicts are often simpler; hence, direct prompting is already strong for some models. Nevertheless, QXR remains competitive on both benchmarks and yields meaningful recall improvements. Hence, QXR 's gains are primarily driven by *preserving* more coherent claims rather than more aggressive filtering.

Model	Direct			QXR		
	P	R	F1	P	R	F1
Qwen2.5 7B	.883	.639	.741	**.902**	**.743**	**.815**
Qwen2.5 32B	.852	.839	.845	**.854**	**.998**	**.920**
Haiku 4.5	**.979**	.575	.724	.925	**.810**	**.863**
Sonnet 4.5	**.962**	.814	**.882**	.887	**.854**	.870
DS-v4-Pro	**.905**	.822	.861	.860	**.952**	**.904**

Table 3: RAMDocs consistency repair results.

5.3 Application: RAG

We now consider a more realistic claim aggregation setting. To do this, we use RAMDocs (Wang et al., 2025), a RAG benchmark containing queries with conflicting retrieved evidence. We evaluate QXR using Qwen2.5 (Qwen et al., 2025), DeepSeek-v4-Pro (DS-v4-Pro) (DeepSeek-AI et al., 2026), Claude Sonnet 4.5 (Anthropic, 2025d) and Claude Haiku 4.5 (Anthropic, 2025c). Each example consists of a query q and retrieved snippets $\mathcal{D} = \{d_1, \ldots, d_n\}$. We convert retrieved snippets into a claim set $F = \{f_1, \ldots, f_N\}$ via deterministic chunking and retain examples with annotated conflicts. Gold conflicts are represented by $\mathcal{U}_{\text{gold}}$, where each $U \in \mathcal{U}_{\text{gold}}$ contains snippets that provide incompatible answers or mutually inconsistent evidence for the same query q. For our experiments $N = 30$ with $|\mathcal{U}_{\text{gold}}| = 10$.

Table 3 shows a similar trend to VitaminC and FEVER, where iterative conflict localization reduces the tendency observed in direct judging to *over-prune* retrieved evidence snippets, thus allowing QXR to preserve more useful information while removing contradictory claims. Furthermore, the results on the 7B and 32B models suggest that structured conflict localization may help compensate for weaker direct set-level reasoning. The results on RAMDocs further suggest that structured diagnosis-and-repair can provide practical gains in realistic RAG settings beyond controlled fact-checking benchmarks.

6 Discussion

6.1 Experimental Observations

When Does Structured Repair Help? Our results suggest that QXR is most useful when a claim set is largely coherent but contains localized conflicts, making unnecessary deletion costly. In such settings, direct judging may correctly detect incon-

sistency while still producing a poor repair. The smaller gains on FEVER also suggest that structured repair need not be preferable when direct judging is already reliable. QXR is therefore best viewed as a preservation-oriented alternative when retaining unaffected information justifies additional oracle calls.

Detection $\not\Rightarrow$ Precise Localization. Our results also highlight a distinction between *detecting* inconsistency and *localizing* its source. Direct judging often maintains high precision but has lower recall, suggesting that a model may recognize that some claims need to be removed while still failing to identify the conflicting claims precisely, resulting in *over-pruning*. The improvements from QXR therefore suggest that consistency detection alone does not provide a complete picture of a model's ability to reason over conflicting claim sets. This also appears to persist under the stronger prompting strategies in Appendix F. Hence, conflict detection and localization should be considered separately.

Global Consistency as an NLP Objective. Taken together, these results suggest that global consistency is not fully captured by existing local verification pipelines. In the contexts of RAG, multi-document summarization, and report generation, the difficulty is not limited to merely recognizing whether there is a contradiction; rather, it includes the challenge of re-establishing coherence without losing valid information. QXR is not only useful in its ability to detect contradictions but also in its framing of global consistency as a *diagnosis and repair* task on sets of natural language claims.

6.2 QXR Extensions

Weighted Claims and Repair. The minimum-deletion objective studied in this work assigns equal cost to removing every claim. In practice, however, some claims might have lower confidence, come from less reliable sources, be outdated, or be less important. A natural generalization assigns each claim $f \in F$ a non-negative deletion cost $w(f) \geq 0$, and replaces the minimum-cardinality hitting set with a minimum-weight hitting set:

$$\min_{H \subseteq F} \sum_{f \in H} w(f) \quad \text{s.t. } H \cap U \neq \varnothing,\ \forall U \in \mathcal{U}.$$

The special case where $w(f) = 1$ for all $f \in F$ is the objective considered in this work. This weighted formulation allows QXR to prefer removing lower-confidence or less reliable claims when multiple repairs can restore consistency. Furthermore, this is important when conflicts overlap, where different minimum-cardinality repairs may remove the same number of claims but preserve information of different value.

Graded Oracle. Real-world consistency judgments, especially from LLMs, may themselves be uncertain rather than binary. One can modify the setup such that the oracle instead returns an inconsistency score $p_{\text{incons}}(S) \in [0, 1]$ for a claim subset S. QXR could then apply a threshold τ, while abstaining or re-querying near the decision boundary. This preserves the diagnosis-and-repair architecture while making uncertainty explicit.

Weighted repair and graded judging address complementary questions: *which claims should be preferred* when repairing a localized conflict, and *how certain is the judge* that a conflict exists? A further extension could combine both, allowing uncertainty in conflict judgments to inform the subsequent repair.

Approximate Repair. QXR currently localizes a subset-minimal conflict before repair. Both diagnosis and repair can be relaxed to trade preservation quality for fewer oracle calls. Localization can terminate early and return a non-minimal conflict; repair can operate on partial conflict families rather than re-diagnosing to a fixpoint; and scopes can be sampled rather than exhaustively swept, reducing the m factor in $Q = \mathcal{O}(I \cdot m \cdot k \log N)$. In the uncached case, QXR 's token cost is at most $\mathcal{O}(Q)$ times that of a single full-set judgment, although localization queries typically contain fewer claims.

7 Conclusion

We introduced global consistency repair for natural-language claim sets: the task of preserving the largest jointly coherent subset of claims while localizing compact sources of inconsistency. This setting captures a failure mode missed by local factuality verification and pairwise contradiction detection, since individually plausible claims may still be globally inconsistent. We instantiated this formulation with QXR, a diagnosis-and-repair framework that uses LLM consistency judgments to localize conflicting subsets and restore coherence through selective repair. Our analysis shows why set-level reasoning is necessary, and our experiments show that QXR preserves substantially more valid claims than direct all-at-once judging across factuality and RAG settings. More broadly, these results suggest that factuality in claim aggregation systems should be treated as a preservation-oriented repair problem, not only as local verification.

Limitations

Our theoretical analysis uses an idealized abstraction of consistency judging. Assumptions such as monotonicity and perfect judging are used to derive worst-case guarantees, but real LLM judgments may be noisy, prompt-sensitive, or non-monotonic. Our experiments therefore use a single LLM call per query ($r = 1$) rather than repeated querying or majority voting. Robustness may benefit from using diverse LLM judges, which we leave to future work.

QXR also does not avoid the worst-case complexity of global repair: a claim set may contain exponentially many MUSes, and minimum-deletion repair reduces to a hitting-set problem. Structured localization therefore trades additional inference cost for better preservation, which may be expensive for large claim sets. Approximate localization or partial repair may be preferable when inference budget is limited. Finally, our experiments use moderate-sized claim collections constructed from existing benchmarks. Larger and more heterogeneous settings, such as long-context RAG and multi-document synthesis, may contain more varied conflicts and will require benchmarks that better reflect real-world claim collections.

Ethical Considerations

Potential Risks. QXR is intended to improve the consistency of claim sets; however, consistency should not be equated with truth. In practice, inconsistencies may reflect legitimate disagreement between sources or varying levels of specificity. Automatically removing claims may therefore suppress valid or minority evidence. This risk also depends on the oracle, which in our setting is an LLM and may inherit model errors or biases.

Use of Generative AI. Generative AI tools were used for writing assistance to improve flow, clarity, and reduce repetition, as well as for coding assistance. The authors developed the scientific ideas, technical content, experimental design, and analysis, and reviewed and revised the resulting text.

References

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.

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

Anthropic. 2025c. Introducing claude haiku 4.5. https://www.anthropic.com/news/claude-haiku-4-5.

Anthropic. 2025d. Introducing claude sonnet 4.5. https://www.anthropic.com/news/claude-sonnet-4-5.

Anthropic. 2026. Introducing claude opus 4.6. https://www.anthropic.com/news/claude-opus-4-6.

Han Cao, Lingwei Wei, Wei Zhou, and Songlin Hu. 2025. Enhancing multi-hop fact verification with structured knowledge-augmented large language models. In *Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence and Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence and Fifteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'25/IAAI'25/EAAI'25. AAAI Press.

Jifan Chen, Grace Kim, Aniruddh Sriram, Greg Durrett, and Eunsol Choi. 2024. Complex claim verification with evidence retrieved in the wild. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3569–3587, Mexico City, Mexico. 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. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14976–14995, Vienna, Austria. Association for Computational Linguistics.

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, Anyi Xu, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chenchen Ling, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chengyu Hou, Chenhao Xu, Chenze Shao, Chong Ruan, Conner Sun, and 300 others. 2026. Deepseek-v4: Towards highly efficient million-token context intelligence. *Preprint*, arXiv:2606.19348.

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*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, and 175 others. 2025. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645:633–638.

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

Michael Hahn and Mark Rofin. 2024. Why are sensitive functions hard for transformers? In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14973–15008, Bangkok, Thailand. 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.

Jaehun Jung, Lianhui Qin, Sean Welleck, Faeze Brahman, Chandra Bhagavatula, Ronan Le Bras, and Yejin Choi. 2022. Maieutic prompting: Logically consistent reasoning with recursive explanations. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1266–1279, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Ulrich Junker. 2004. Quickxplain: preferred explanations and relaxations for over-constrained problems. In *Proceedings of the 19th National Conference on Artificial Intelligence*, AAAI'04, page 167–172. AAAI Press.

Nora Kassner, Oyvind Tafjord, Hinrich Schütze, and Peter Clark. 2021. BeliefBank: Adding memory to a pre-trained language model for a systematic notion of belief. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8849–8861, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

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.

Mistral AI team. 2024. Large enough. `https://mistral.ai/news/mistral-large-2407/`.

Eric Mitchell, Joseph Noh, Siyan Li, Will Armstrong, Ananth Agarwal, Patrick Liu, Chelsea Finn, and Christopher Manning. 2022. Enhancing self-consistency and performance of pre-trained language models through natural language inference. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1754–1768, Abu Dhabi, United Arab Emirates. 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.

Qwen, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, and 24 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

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.

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

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.

Mooho Song, Hye Ryung Son, and Jay-Yoon Lee. 2025. Introducing verification task of set consistency with set-consistency energy networks. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 33346–33366, Vienna, Austria. 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, 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*. Springer.

Han Wang, Archiki Prasad, Elias Stengel-Eskin, and Mohit Bansal. 2025. Retrieval-augmented generation with conflicting evidence. In *Second Conference on Language Modeling*.

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.

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*.

A Proofs and Theoretical Details

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

Theorem A.1 (Objective Equivalence)**.** *For all $i \in [m]$, maximizing coverage,*

$$\max_{F' \subseteq F} |F'| \quad \textit{s.t.} \quad A(F' \cap C_i) = \mathsf{cons}, \tag{1}$$

is equivalent to minimizing deletions,

$$\min_{R \subseteq F} |R| \quad \textit{s.t.} \quad A((F \setminus R) \cap C_i) = \mathsf{cons} \tag{2}$$

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

$$|F'| = |F| - |R|. \tag{3}$$

Hence, maximizing $|F'|$ is equivalent to minimizing $|R|$. Under this bijection, the feasibility constraints in Equations (1) and (2) are identical. □

A.2 Reduction to Hitting Set

Let $\mathcal{A}_{\mathsf{incons}}$ be the family of all MUSes contained in the scopes C. Formally,

$$\mathcal{A}_{\mathsf{incons}} = \{\, U \subseteq C_j : j \in [m],\ U \text{ is a MUS w.r.t. } A \,\}. \tag{4}$$

Theorem A.2 (Repair-Hitting Set Equivalence)**.** *A set $R \subseteq F$ is feasible for Equation (2) if and only if it is a hitting set for $\mathcal{A}_{\mathsf{incons}}$.*

Proof. First, assume $R \subseteq F$ is feasible for the minimum-deletion problem. Suppose for contradiction that 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

$$U \subseteq (F \setminus R) \cap C_i. \tag{5}$$

Since $A(U) = \mathsf{incons}$, monotonicity implies that $(F \setminus R) \cap C_i$ is inconsistent, contradicting feasibility of R. Hence $R \cap U \neq \varnothing$ for all $U \in \mathcal{A}_{\mathsf{incons}}$, so R is a hitting set.

Conversely, assume R is a hitting set for $\mathcal{A}_{\mathsf{incons}}$, i.e.,

$$R \cap U \neq \varnothing \quad \forall U \in \mathcal{A}_{\mathsf{incons}}. \tag{6}$$

Suppose for contradiction that R is not feasible. Then there exists some i such that

$$A((F \setminus R) \cap C_i) = \mathsf{incons}. \tag{7}$$

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, so R is feasible. □

A.3 Pairwise Checks Are Insufficient

Theorem A.3 (Pairwise Insufficiency)**.** *For any $N \geq 3$, there exists a fact set F of size $\binom{N}{2}$ such that every pair of facts is jointly consistent, yet F is globally inconsistent.*

Proof. Consider a set of variables $x_1, \ldots, x_N$ taking values in a finite domain D of labels. We build a constraint graph $G = (V, E)$ where $V = \{1, \ldots, N\}$ and an edge $(i, j) \in E$ represents a pairwise incompatibility constraint $x_i \neq x_j$. A labeling $\ell : V \to D$ is consistent if and only if it is a proper coloring of G, meaning adjacent vertices receive different labels. Thus, global consistency of the fact set corresponds to k-colorability of G, where $k = |D|$.

Let G be the complete graph K_N on N vertices, and let the domain have size $|D| = N - 1$. For each edge (i, j), include the pairwise constraint $x_i \neq x_j$ as a fact.

We first show that every pair of facts is jointly consistent. Any two facts correspond to two edges in K_N. The subgraph induced by two distinct edges is either two disjoint edges or two edges sharing one endpoint. In all cases, this subgraph is bipartite and hence 2-colorable. Since $|D| = N - 1 \geq 2$, there is an assignment of labels satisfying both facts. Therefore every pair of facts in F is jointly consistent. However, a proper coloring of K_N requires N distinct labels because $\chi(K_N) = N$. Since $|D| = N - 1$, the complete graph K_N is not $(N-1)$-colorable. Equivalently, by the pigeonhole principle, N vertices cannot be assigned pairwise distinct labels from a set of only $N - 1$ labels. Therefore, the full set of facts is inconsistent.

Hence, for every unordered pair $i < j$, include the claim $f_{ij} = x_i \neq x_j$. Since there are $\binom{N}{2}$ unordered pairs, the resulting claim set has size $|F| = \binom{N}{2}$. □

For intuition, consider three Boolean variables $A, B, C \in \{0, 1\}$ and the three facts $f_1 : A \oplus B = 1$, $f_2 : B \oplus C = 1$, and $f_3 : C \oplus A = 1$. Any two facts are jointly satisfiable, for example f_1, f_2 hold for $A = 1, B = 0, C = 1$. But all three cannot hold simultaneously: from f_1 and f_2 we have $A = \neg B$ and $C = \neg B$, hence $A = C$, contradicting f_3 since then $C \oplus A = 0$. This is the special case $N = 3$ of the graph-based construction.

A.4 Soundness under a Perfect Oracle

Theorem A.4 (Soundness under Perfect Oracle)**.** *Assume that the LLM oracle O has zero error and matches the ground-truth consistency function A. Let $R \subseteq F$ be the set of facts removed by* QXR. *Then, for every $i \in [m]$, the retained subset $F' = F \setminus R$ satisfies*

$$A(F' \cap C_i) = \mathsf{cons}. \tag{8}$$

Proof. By lines 11-12 of Algorithm 1, after every iteration each stored scope descended from C_i is $C_i \cap F'$ where F' is the current retained fact set. When the algorithm terminates, the condition $\exists\ C_i \in C$ s.t. $O(C_i) = \mathsf{incons}$ evaluates to false, hence $O(F' \cap C_i) = \mathsf{cons}$ for every $i \in [m]$. Since the oracle is perfect, we therefore have $O \equiv A$ and therefore $A(F' \cap C_i) = \mathsf{cons}$ for every $i \in [m]$. □

A.5 Soundness with Respect to Extracted Conflicts

In practical scenarios, the MUSes $\mathcal{U}_{\text{found}}$ extracted by Algorithm 1 may contain errors because the procedure depends on a noisy oracle O. However, the retained set is still sound with respect to the conflicts actually extracted.

Theorem A.5 (Practical Soundness w.r.t. Extracted Conflicts)**.** *Let $\mathcal{U}_{\text{found}}$ be the set of conflict sets extracted by Algorithm 1 under oracle O. Let $R \subseteq F$ satisfy*

$$R \cap U \neq \varnothing \quad \forall U \in \mathcal{U}_{\text{found}}. \tag{9}$$

Then the retained set $F \setminus R$ contains none of the extracted conflicts in $\mathcal{U}_{\text{found}}$.

Proof. For every $U \in \mathcal{U}_{\text{found}}$, the condition $R \cap U \neq \varnothing$ implies that $U \not\subseteq F \setminus R$. Thus, none of the discovered conflict sets in $\mathcal{U}_{\text{found}}$ remain after removal. □

A.6 Greedy Hitting Set Approximation

Lemma A.1 (Greedy Hitting Set Approximation (Vazirani, 2001))**.** *Let $\mathcal{U}$ be a family of $m = |\mathcal{U}|$ MUSes, or conflict sets, over facts F. The greedy hitting-set algorithm returns a hitting set H satisfying*

$$|H| \leq (1 + \ln m)\,|H^\star|, \tag{10}$$

where $H^\star$ is an optimal minimum-cardinality hitting set of $\mathcal{U}$.

Proof. The hitting-set problem is dual to set cover. For each fact $f \in F$, define the set

$$S_f = \{\, U \in \mathcal{U} : f \in U \,\}. \tag{11}$$

Choosing facts that hit every conflict set is equivalent to choosing sets S_f that cover every element of $\mathcal{U}$. The standard greedy set-cover analysis gives an approximation factor $H_m \leq 1 + \ln m$, where $m = |\mathcal{U}|$. Translating back to hitting set gives

$$|H| \leq (1 + \ln m)\,|H^\star|. \tag{12}$$

□

A.7 Information-Theoretic Cost of Conflict Localization

To understand the intrinsic cost of identifying conflicting claims, we isolate the localization step from the repair problem and consider a simplified setting with a single hidden conflict.

Definition A.6 (Hidden-Conflict Query Model)**.** Let F be a fact set with $|F| = N$, and let $U^\star \subseteq F$ be an unknown conflict set of size $|U^\star| = k$. A subset query on $S \subseteq F$ returns

$$O^\star(S) = \begin{cases} \textsf{incons} & \text{if } U^\star \subseteq S, \\ \textsf{cons} & \text{otherwise.} \end{cases} \tag{13}$$

The goal is to identify $U^\star$ exactly using adaptive subset queries.

Theorem A.7 (Lower Bound for Exact Conflict Localization)**.** *In the hidden-conflict query model, any adaptive algorithm that always identifies $U^\star$ must make at least $\left\lceil \log_2 \binom{N}{k} \right\rceil$ queries in the worst case.*

Proof. There are exactly $\binom{N}{k}$ possible choices for the hidden conflict $U^\star$. Any deterministic adaptive algorithm can be viewed as a binary decision tree whose internal nodes are oracle queries and whose outgoing edges correspond to the two possible oracle responses, cons and incons. If the algorithm makes at most q queries in the worst case, then this decision tree has depth at most q and therefore at most 2^q leaves.

For the algorithm to identify $U^\star$ exactly for every possible hidden conflict, each of the $\binom{N}{k}$ possibilities must lead to a distinct leaf. Hence $2^q \geq \binom{N}{k}$. Taking logarithms gives $q \geq \log_2 \binom{N}{k}$. Since q is an integer, the worst-case number of queries must be at least $\left\lceil \log_2 \binom{N}{k} \right\rceil$. □

A.8 Cost of Exact Localization

Corollary A.8 (Cost of Exact Localization)**.** *For $1 \leq k \leq N/2$,*

$$\log \binom{N}{k} = \Theta(k \log(N/k))\,. \tag{14}$$

Hence any exact deterministic adaptive subset-query algorithm requires $\Omega(k \log(N/k))$ queries in the worst case. The QuickXplain instantiation used in QXR *localizes a conflict of size k using $\mathcal{O}(k \log N)$ consistency queries, matching this lower bound up to logarithmic factors and becoming particularly sharp in the small-conflict regime.*

Proof. For $1 \leq k \leq N/2$, standard binomial coefficient bounds give

$$\left(\frac{N}{k}\right)^k \leq \binom{N}{k} \leq \left(\frac{eN}{k}\right)^k. \tag{15}$$

Taking logarithms yields

$$k \log(N/k) \leq \log \binom{N}{k} \leq k \log(eN/k). \tag{16}$$

Thus $\log \binom{N}{k} = \Theta(k \log(N/k))$. Combining this with Theorem A.7 gives the stated lower bound. □

A.9 Query Complexity of QXR

Theorem A.9 (Query Complexity of Algorithm 1)**.** *Let $N = |F|$ be the number of claims, $m = |C|$ the number of scopes, k the maximum size of any MUS localized by* QXR*, and I the number of outer iterations until termination. Assuming* LOCALIZEMUS *is instantiated with QuickXplain, Algorithm 1 makes at most*

$$\mathcal{O}(I \cdot m \cdot (1 + k \log N)) \tag{17}$$

consistency queries. For $k \geq 1$, this simplifies to

$$\mathcal{O}(I \cdot m \cdot k \log N)\,. \tag{18}$$

Proof. At the start of each outer iteration, QXR checks each of the m scopes to identify the currently inconsistent scopes, incurring $\mathcal{O}(m)$ oracle calls. For each inconsistent scope, it invokes LOCALIZEMUS. Under the QuickXplain instantiation, localizing a MUS of size at most k within a scope of size at most N requires $\mathcal{O}(k \log N)$ consistency queries. Since at most m scopes are localized in a single outer iteration, each iteration requires

$$\mathcal{O}(m + mk \log N) = \mathcal{O}(m(1 + k \log N)) \tag{19}$$

oracle calls. Over I outer iterations, the total number of calls is

$$\mathcal{O}(I \cdot m \cdot (1 + k \log N))\,. \tag{20}$$

For $k \geq 1$, the term $1 + k \log N$ is absorbed into $\mathcal{O}(k \log N)$, yielding $\mathcal{O}(I \cdot m \cdot k \log N)$. □

B QuickXplain Pseudocode

QuickXplain (Junker, 2004) as shown in Algorithm 2, localizes a minimal unsatisfiable subset from an inconsistent set S under a consistency oracle $O : 2^F \to \{\mathsf{cons}, \mathsf{incons}\}$. The algorithm adaptively queries subsets to find a subset-minimal inconsistent core with logarithmic depth in $|S|$. QuickXplain narrows the inconsistent subset by recursively testing halves of S while conditioning on a background context B. For a perfect oracle, QuickXplain requires $\mathcal{O}(k \log |S|)$ queries to locate a conflict of size k.

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 $O(B) = \mathsf{incons}$ **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 $\Delta_1 \cup \Delta_2$

C Additional Structural Theory

The main text focuses on the algorithmic behavior of QXR under adaptive localization and greedy repair. The results in this section address a complementary question: when is exact global consistency repair structurally tractable, independent of the particular QXR implementation? Worst-case hardness alone does not explain why decomposition-based methods can work well in practice. In many NLP settings, scopes overlap in structured rather than arbitrary ways: for example, claims may be grouped by document, entity, event, retrieved evidence cluster, or local subtask. The scope interaction hypergraph formalizes this overlap structure.

The join-tree results below identify a favorable regime: if scope overlaps are tree-like and global feasibility is captured by scope-wise feasibility, exact repair can be solved by dynamic programming. We also further show that when the conflict graph has bounded treewidth, exact repair is tractable with complexity controlled by that treewidth rather than by the full number of facts. We do not claim that open-ended NLP instances satisfy these assumptions exactly. Rather, they provide a useful lens for understanding why decomposition can be effective when conflicts are localized and scope overlaps are limited.

C.1 Fact-Disjoint Decomposition

Definition C.1 (Scope Interaction Hypergraph)**.** Let F be the fact set and $C = \{C_1, \ldots, C_m\}$ a family of scopes with $C_i \subseteq F$. The *scope interaction hypergraph* is the hypergraph $\mathcal{H} = (F, C)$ whose vertices are facts and whose hyperedges are scopes.

Lemma C.1 (Additive Decomposition over Fact-Disjoint Components)**.** *If the scope interaction hypergraph $\mathcal{H}$ decomposes into fact-disjoint connected components, then the minimum-deletion repair problem decomposes additively across components.*

Proof. Let the fact-disjoint connected components of $\mathcal{H}$ be

$$(F^{(1)}, C^{(1)}), \ldots, (F^{(t)}, C^{(t)}), \tag{21}$$

where $F = \dot{\bigcup}_{\ell=1}^{t} F^{(\ell)}$ and every scope in $C^{(\ell)}$ is contained in $F^{(\ell)}$. Here $\dot{\cup}$ denotes disjoint union.

For any feasible global repair $R \subseteq F$, define $R^{(\ell)} = R \cap F^{(\ell)}$. Since each scope $C_i \in C^{(\ell)}$ lies entirely inside $F^{(\ell)}$, we have

$$(F \setminus R) \cap C_i = (F^{(\ell)} \setminus R^{(\ell)}) \cap C_i. \tag{22}$$

Hence each $R^{(\ell)}$ is feasible for the repair subproblem on component ℓ. Therefore $|R^{(\ell)}| \geq \mathrm{OPT}_\ell$ for every ℓ, where OPT_ℓ denotes the minimum deletion cost in component ℓ. Summing over components gives

$$|R| = \sum_{\ell=1}^{t} |R^{(\ell)}| \geq \sum_{\ell=1}^{t} \mathrm{OPT}_\ell. \tag{23}$$

Conversely, let $R^\star_\ell$ be an optimal repair for component ℓ, and define $R^\star \triangleq \bigcup_{\ell=1}^{t} R^\star_\ell$. Again using that scopes do not cross components, every scope $C_i \in C^{(\ell)}$ satisfies

$$(F \setminus R^\star) \cap C_i = (F^{(\ell)} \setminus R^\star_\ell) \cap C_i, \tag{24}$$

which is consistent by feasibility of $R^\star_\ell$. Thus $R^\star$ is a feasible global repair, with

$$|R^\star| = \sum_{\ell=1}^{t} |R^\star_\ell| = \sum_{\ell=1}^{t} \mathrm{OPT}_\ell. \tag{25}$$

Therefore the global optimum decomposes additively across connected components. □

C.2 Scope Subsumption

The scoped formulation generalizes global consistency, recovered by $C = \{F\}$. Under monotonicity, nested scopes are redundant: if one scope is contained in another, satisfying the larger scope already guarantees satisfaction of the smaller one. Thus, when $F \in C$, all proper subscopes can be removed without changing the feasible repair set. Scope-locality should therefore be understood as an assumption on the remaining non-redundant scope family.

Lemma C.2 (Scope Subsumption)**.** *Assume inconsistency is monotone. If $C_j \subseteq C_i$, then the feasibility constraint on C_j is implied by the feasibility constraint on C_i, so C_j may be removed from the scope family without changing the feasible set of repairs.*

Proof. Since $C_j \subseteq C_i$, we have

$$(F \setminus R) \cap C_j \subseteq (F \setminus R) \cap C_i. \tag{26}$$

Suppose for contradiction that $A((F \setminus R) \cap C_j) = \mathsf{incons}$. By monotonicity of inconsistency, this implies

$$A((F \setminus R) \cap C_i) = \mathsf{incons}, \tag{27}$$

contradicting feasibility of C_i. Therefore feasibility on C_i implies feasibility on C_j. □

C.3 Exact Repair on Join Trees

Definition C.2 (Join-Tree Acyclicity)**.** The scope interaction hypergraph $\mathcal{H} = (F, C)$ is *join-tree acyclic* if there exists a tree T whose nodes are the scopes $C_i \in C$ such that, for every fact $f \in F$, the set of scopes containing f induces a connected subtree of T:

$$\{\, C_i \in C : f \in C_i \,\}. \tag{28}$$

Definition C.3 (Scope Locality)**.** We say the repair problem is *scope-local* if global feasibility is fully captured by scope-wise feasibility. Equivalently, every global inconsistency contains a minimal unsatisfiable subset fully contained in some scope C_i.

Join-tree acyclicity is useful because it gives a principled version of "structured decomposition." The connected-subtree condition ensures that if a fact appears in multiple scopes, all occurrences of that fact can be coordinated through a separator in the tree. This allows dynamic programming to combine locally feasible repairs without making inconsistent keep/delete decisions for shared facts. Scope-locality is not required for the dynamic program to optimize the scope-wise objective in Equation (2); rather, it ensures that feasibility under these scope-wise constraints coincides with global consistency.

Theorem C.4 (Exact Repair on Join Trees)**.** *Assume:*

1. *the repair problem is scope-local;*
2. *the scope interaction hypergraph $\mathcal{H}$ is join-tree acyclic.*

Then the minimum-deletion repair problem

$$\min_{R \subseteq F} |R| \quad \textit{s.t.} \quad A((F \setminus R) \cap C_i) = \mathsf{cons}, \tag{29}$$

$\forall i \in [m]$, is solvable exactly by dynamic programming over a join tree of $\mathcal{H}$.

Proof. Let T be a join tree of $\mathcal{H}$ whose nodes are the scopes C_i. Root T at an arbitrary scope C_r. For each non-root node i, let $\mathrm{Pa}(i)$ denote its parent and let $S_i \triangleq C_i \cap C_{\mathrm{Pa}(i)}$ be the separator between i and its parent. For the root, define $S_r = \varnothing$.

For each node i, let T_i denote the subtree rooted at i, and let $F_i^{\downarrow} \triangleq \bigcup_{C_j \in T_i} C_j$ be the set of facts appearing in that subtree.

We define a dynamic program over T. A state at node i is a keep/delete assignment σ on the separator S_i, specifying which facts in S_i are retained. Let $\mathrm{DP}_i(\sigma)$ denote the minimum number of deletions inside $F_i^{\downarrow} \setminus S_i$ such that:

1. the resulting retained facts in the subtree agree with σ on S_i;
2. every scope in the subtree is locally consistent; and
3. the retained sets chosen for adjacent scopes agree on their overlaps.

Fix a node i and a separator assignment σ on S_i. Any feasible solution for the subtree rooted at i must choose a retained subset on the facts of C_i that extends σ. Equivalently, it chooses a keep-/delete assignment τ on C_i whose restriction to S_i equals σ, and such that the retained subset of C_i is locally consistent. For each child j of i, this assignment induces a separator assignment $\tau|_{C_i \cap C_j}$ for the child subtree. Hence the optimal value satisfies $\mathrm{DP}_i(\sigma) = \min_{\substack{\tau \text{ on } C_i \\ \tau|_{S_i}=\sigma \\ \tau \text{ feasible}}} \{\kappa\}$, where

$$\kappa = \mathrm{cost}_i(\tau) + \sum_{j \in \mathrm{child}(i)} \mathrm{DP}_j\big(\tau|_{C_i \cap C_j}\big) \quad (30)$$

and $\mathrm{cost}_i(\tau)$ counts deletions in $C_i \setminus S_i$.

We now argue correctness. First, any solution assembled by the recurrence is globally feasible. By construction, each scope is locally feasible, and adjacent scopes agree on shared facts. Since $\mathcal{H}$ is join-tree acyclic, the scopes containing any fact form a connected subtree, so these pairwise agreements define a unique global keep/delete decision for every fact. Therefore the dynamic program constructs a well-defined retained set $X \subseteq F$. Because the problem is scope-local, local feasibility of every scope implies global feasibility of X.

Conversely, let $X^\star \subseteq F$ be any globally feasible retained set. It induces, for every node i, a separator assignment on S_i and a local keep/delete assignment on C_i. Since $X^\star$ is globally feasible and inconsistency is monotone, $X^* \cap C_i \subseteq X^*$ implies every scope is locally feasible. These induced assignments therefore satisfy the dynamic-program constraints and achieve exactly the deletion cost of $X^\star$. Hence every globally feasible repair is represented by some valid collection of DP choices.

Thus the dynamic program is both sound and complete, and its optimum value equals the exact minimum-deletion repair cost. □

C.4 Tractability under Bounded Conflict-Graph Treewidth

The complexity of exact repair could also be characterized by the interaction structure among minimal conflicts. Let $\mathcal{A}_{\text{incons}}$ be as defined in Equation (4) (the family of scope-contained MUSes).

Definition C.5 (Tree Decomposition and Treewidth)**.** Let $G = (V, E)$ be a graph. A *tree decomposition* of G is a tree T whose nodes $x \in V(T)$ are associated with bags $B_x \subseteq V$, such that every vertex and edge of G is contained in some bag and, for every $v \in V$, the bags containing v form a connected subtree of T. The width of the decomposition is $\max_{x \in V(T)} |B_x| - 1$, and the *treewidth* $\mathrm{tw}(G)$ is the minimum width over all tree decompositions of G.

Definition C.6 (Conflict Graph)**.** The *conflict graph* induced by $\mathcal{A}_{\text{incons}}$ is $G_{\mathcal{A}} = (F, E)$ where $\{f, f'\} \in E \iff (f \neq f' \wedge \exists U \in \mathcal{A}_{\text{incons}}$ such that $f, f' \in U)$.

Under this definition, every conflict $U \in \mathcal{A}_{\text{incons}}$ forms a clique in $G_{\mathcal{A}}$. Figure 2 illustrates the above concepts.

Theorem C.7 (Exact Repair under Bounded Conflict-Graph Treewidth)**.** *Suppose we are given $\mathcal{A}_{\text{incons}}$, and let $t = \mathrm{tw}(G_{\mathcal{A}})$ denote the treewidth of $G_{\mathcal{A}}$. Given a tree decomposition of $G_{\mathcal{A}}$ of width t, the minimum-deletion repair problem in Equation (2) can be solved exactly in $2^{t+1} \cdot \mathrm{poly}(|F| + |\mathcal{A}_{\text{incons}}|)$ time.*

Proof. By Theorem A.2, a deletion set $R \subseteq F$ is feasible for Equation (2) if and only if $R \cap U \neq \varnothing$ for every $U \in \mathcal{A}_{\text{incons}}$. Therefore, exact minimum-deletion repair is equivalent to minimum hitting set over $\mathcal{A}_{\text{incons}}$.

First, we show every conflict lies within a single bag. Under the definition of $G_{\mathcal{A}}$, there exists an edge between every pair of distinct elements of U, therefore, a MUS U induces a clique. Now, suppose U forms a $|U|$-clique, for each vertex $f_i \in U$, consider the set of tree nodes whose bags contain it: $T_i = \{x \in V(T) \mid f_i \in B_x\}$. By the running-intersection property, each T_i is a connected subtree of T. Because U is a $|U|$-clique, every pair of distinct vertices $f_i, f_j \in U$ satisfies $\{f_i, f_j\} \in E(G_{\mathcal{A}})$. By edge coverage, some bag contains both f_i and f_j, and hence $T_i \cap T_j \neq \varnothing$. Thus, the subtrees $T_1, \ldots, T_{|U|}$ are pairwise intersecting and subtrees of a tree satisfy the Helly property and hence $\bigcap_{i=1}^{|U|} T_i \neq \varnothing$. Therefore, there exists some $x \in V(T)$ such that $U \subseteq B_x$. Assign each $U \in \mathcal{A}_{\text{incons}}$ to one such bag.

We now perform dynamic programming over the given bag-tree decomposition. We may assume the decomposition is a nice tree decomposition of the same width and polynomial size. For each bag B_x, a state specifies the keep/delete assignment of the facts in B_x. Since $|B_x| \leq t + 1$, there are at most 2^{t+1} states per bag. A state is only feasible

if it deletes at least one fact from every conflict assigned to that bag. Transitions between adjacent bags enforce agreement of their shared facts and accumulate deletion costs without double counting.

The minimum-cost feasible state at the root therefore yields a minimum hitting set of $\mathcal{A}_{\text{incons}}$ and by Theorem A.2, an exact minimum-deletion repair. Therefore, our dynamic program has at most 2^{t+1} states per bag, with polynomial time for additional checks per bag, giving total running time $2^{t+1} \cdot \text{poly}(|F| + |\mathcal{A}_{\text{incons}}|)$. □

Taken together, these structural results identify a favorable regime for exact repair. Lemma C.1 shows that independent components can be solved separately, while Theorems C.4 and C.7 show that sparse, tree-like conflict interactions admit dynamic programming with complexity controlled by conflict-graph treewidth rather than the full number of facts. Thus, when inconsistencies are localized and conflict interactions are limited, global consistency repair need not exhibit worst-case combinatorial behavior.

The framework also extends directly to weighted repair as mentioned in Section 6.2. In many NLP applications, facts carry non-uniform importance due to confidence, provenance, or task-specific utility. Assigning each fact a deletion cost and minimizing total removed weight rather than cardinality preserves the same connection to weighted hitting set, and the join-tree dynamic program carries over by replacing deletion counts with deletion weights.

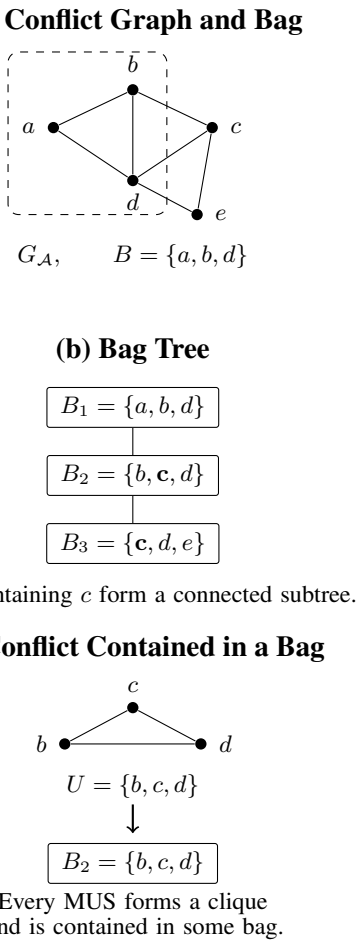

Figure 2: Illustration of bounded conflict-graph treewidth. (a) A bag contains a subset of conflict-graph vertices. (b) Bags form a tree satisfying the running-intersection property. (c) Every MUS induces a clique in $G_{\mathcal{A}}$ and is therefore contained in some bag. Here $\mathcal{A}_{\text{incons}} = \{\{a, b, d\}, \{b, c, d\}, \{c, d, e\}\}$.

D Complexity of Pairwise/NLI Baselines

Methods that decide consistency by evaluating all sentence pairs require $\binom{N}{2}$ oracle calls. For typical fact-set sizes, e.g., $N \in [30, 100]$, that implies 435–4950 calls per instance. Under noisy LLM oracles, each decision may further require r repetitions. In contrast, QXR performs at most $\mathcal{O}(I \cdot m \cdot k \log N)$ queries, where k is the maximum MUS size, m is the number of scopes, and I is the number of outer rounds. In our experiments, $m = 1$ and conflicts are small. Table 4 summarizes such query complexities. We use direct all-at-once LLM judging as the primary baseline because it is the strongest natural black-box alternative to QXR for the same preservation-oriented task. Given the full claim set, the baseline is prompted to return the largest jointly coherent subset, matching the input-output format and objective of QXR. This makes the comparison conservative: both methods rely on the same LLM judge and the same natural-language claim representation, but QXR adds explicit conflict localization and hitting-set repair. Pairwise NLI or contradiction-detection baselines are not directly comparable, since pairwise consistency does not certify global consistency and such methods do not by themselves return a maximal coherent subset. Exhaustive subset enumeration or exact MUS search is combinatorially infeasible at realistic claim-set sizes, while symbolic solvers require a formalization of natural-language claims that is unavailable in the black-box setting. Thus, direct LLM judging is the appropriate primary baseline for testing whether structured diagnosis-and-repair improves over the most natural end-to-end alternative.

E Prompts

All experiments use publicly available LLM APIs and standard benchmark datasets (e.g., VitaminC)

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

Table 4: Common NLI-style baselines for consistency checking and their computational complexity. Pairwise and entailment-graph methods grow quadratically with the number of claims, while multi-premise NLI corresponds to the direct all-at-once baseline.

under their existing licenses. 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).

```
Factual statements. Some may contradict.

{bg}
Statements:
{facts_block}

Respond ONLY:
- CONSISTENT
- INCONSISTENT

Answer:
```

Direct baseline (zero-shot).

```
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).

```
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).

```
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).

```
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).

```
Given the following facts, some may
contradict.
```

	Claude-3.7			Claude-4			DeepSeek-R1			GPT-OSS-120B			Mistral Large 2		
Prompting Style	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Zero-shot	.986	.681	.806	.989	.928	.958	.993	.860	.922	.994	.876	.931	.947	.568	.710
Chain-of-Thought	.973	.684	.803	.987	.908	.946	.996	.850	.917	.995	.851	.917	.997	.570	.725
Decomposition	.987	.626	.766	.677	.608	.641	.996	.882	.936	.954	.816	.880	.991	.535	.695
Few-shot	.975	.658	.786	.992	.891	.939	.882	.761	.817	.997	.896	.944	.238	.142	.178
Self-consistency	.986	.547	.704	.992	.768	.866	.987	.718	.831	.996	.475	.643	.990	.522	.684
Original Direct	.979	.854	.912	.956	.877	.915	.980	.730	.837	.984	.926	.953	.955	.603	.739
QXR	.956	.975	**.965**	.938	.983	**.960**	.973	.990	**.981**	.956	.995	**.975**	.968	.978	**.972**

Table 5: Prompt-sensitivity results across models. We compare direct judging under alternative prompting strategies with the original direct baseline and QXR.

```
Return ONLY a Python list of facts that
are mutually consistent.

Facts:
{facts_block}

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

F Prompting Analysis

All main experiments 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. Results are shown in Table 5. Across 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.