

# ARGUS: Argumentation-Based Minimal-Change Repair for Verifiable LLM Self-Explanations

Paper ID: 607

Anonymous Authors

anonymous@example.com

## Abstract

Large language models produce natural-language rationales for their outputs, yet these explanations are frequently unfaithful to the model’s internal reasoning and lack formal mechanisms for maintenance under evolving evidence. We introduce ARGUS, a framework that structures LLM self-explanations as Dung-style abstract argumentation frameworks, verifies them under grounded and preferred semantics, and—when an evidence update renders the explanation inconsistent—computes a minimum-cost set of edit operations that restores the desired acceptability status of the target argument. The repair operator satisfies adapted AGM revision postulates (success, inclusion, vacuity) and admits a complexity-theoretic characterization: the decision problem is in P under grounded semantics and NP-complete under preferred and stable semantics. A  $k$ -neighborhood approximation and an answer set programming (ASP) encoding ensure scalability to practical framework sizes. We validate the framework on HotpotQA and FEVER, where ARGUS achieves relative improvements of 10.3% in faithfulness and 14.5% in contestability over the strongest argumentation baseline while requiring fewer repair operations than all competing methods.

## 1 Introduction

Large language models generate natural-language explanations for their outputs, yet mounting evidence indicates that these self-explanations are frequently unfaithful to the model’s internal reasoning process. Recent studies demonstrate that LLM rationales can be inconsistent with the computations that actually produce the answer (Ye and Durst 2024), and that chain-of-thought traces are often post-hoc rationalizations rather than faithful accounts of inference (Lanham et al. 2023). The gap between *apparent* and *actual* reasoning makes the verification and maintenance of explanations a central knowledge representation challenge, particularly in domains such as medical diagnosis and legal reasoning where explanation correctness is critical.

As illustrated in Figure 1, current approaches fall short along two complementary dimensions. Self-correction methods (Madaan et al. 2023; Shinn et al. 2023; Gao et al. 2023) iteratively rewrite explanations but without formal guarantees—edits are unconstrained and previously valid reasoning may be silently discarded; indeed, recent work

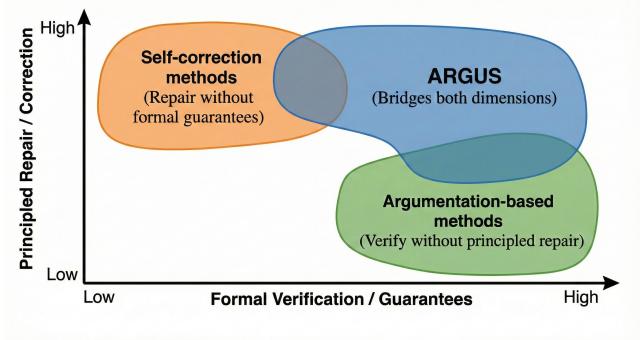


Figure 1: Qualitative positioning of ARGUS. Self-correction methods (orange) repair without formal guarantees; argumentation-based methods (green) verify without principled repair. ARGUS (blue region) bridges both dimensions.

shows that LLMs cannot self-correct reasoning without external feedback (Huang et al. 2024). Argumentation-based approaches (Freedman et al. 2025; Jin et al. 2026) verify explanations against formal semantics but treat verification as terminal: when new evidence arrives, they offer no principled way to update the explanation while preserving consistency. No existing framework provides a formal notion of *minimal change* for maintaining LLM explanations under evolving evidence.

The following example, revisited throughout the paper, illustrates the problem concretely.

**Example 1** (Medical Diagnosis). *A question-answering system is asked to diagnose a patient with fatigue and joint pain. The LLM answers “Lupus” with four argument units:  $a_1$  (“chronic fatigue reported”),  $a_2$  (“polyarthralgia present”),  $a_3$  (“Lupus commonly presents with these symptoms”), and target  $a_4$  (“most likely diagnosis is Lupus”). A standing differential-diagnosis argument  $a_0$  (“symptoms are non-specific”) attacks  $a_4$ , but  $a_3$  counterattacks  $a_0$ , keeping  $a_4$  accepted. A new lab result  $a_5$  (“ANA test is negative”) attacks  $a_3$ , removing the defense of  $a_4$ : the differential  $a_0$  reinstates, rendering  $a_4$  no longer accepted un-*

der grounded semantics. An unconstrained self-correction system might regenerate the entire explanation, discarding the valid units  $a_1$  and  $a_2$ . A minimal-change repair instead seeks the smallest edit—such as introducing  $a_6$  (“anti-dsDNA positive”) attacking  $a_5$ —to restore  $a_4$  at cost 2 (visualized in Figure 2).

We propose ARGUS, a framework that bridges this gap by unifying argumentation-based verification with minimal-change repair. Given an LLM-generated explanation, ARGUS decomposes it into atomic argument units, constructs an argumentation framework in the sense of Dung (Dung 1995), and verifies whether the target claim is accepted under a chosen semantics. When new evidence renders the explanation inconsistent, ARGUS computes a minimum-cost set of edit operations—adding or removing arguments and attacks—that restores the desired acceptability status. The repair operator draws on two classical KR traditions: the AGM theory of belief revision (Alchourrón, Gärdenfors, and Makinson 1985), which supplies the minimal-change principle, and argumentation dynamics (Cayrol, de Saint-Cyr, and Lagasquie-Schiex 2020), which provides formal machinery for structural change. Because the repair operates on an explicit graph structure external to the LLM, it admits formal guarantees—AGM compliance, complexity bounds, provable preservation of unaffected reasoning—that are unattainable when editing model internals or regenerating from scratch.

Our contributions are as follows:

1. **(C1)** A framework that structures LLM self-explanations as Dung-style argumentation frameworks, verifies them under grounded and preferred semantics, and produces defense-set certificates for interpretable verdicts (§4).
2. **(C2)** A minimal-change repair operator that formulates explanation maintenance as a new optimization problem over argumentation frameworks, satisfying adapted AGM revision postulates with a complexity analysis placing the problem in P under grounded semantics and NP-complete under preferred and stable semantics (§4.4–§5).
3. **(C3)** A scalable ASP encoding with a  $k$ -neighborhood approximation that preserves repair quality while substantially reducing solver grounding (§4).
4. **(C4)** An empirical evaluation on HotpotQA and FEVER validating the formal properties and demonstrating improvements in faithfulness, contestability, and repair cost w.r.t. seven baselines (§6).

## 2 Related Work

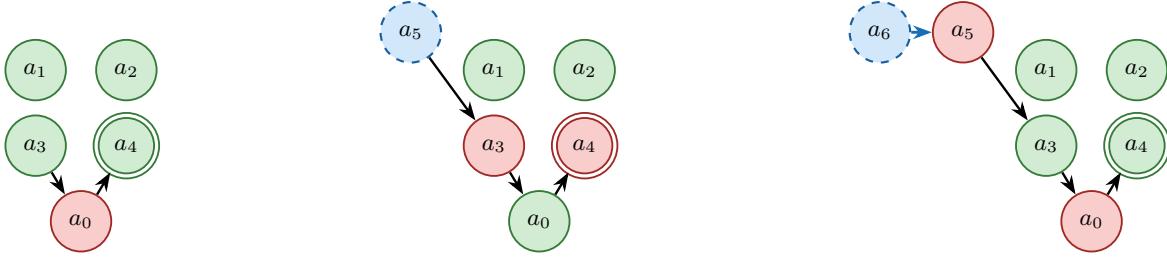
Our work connects three lines of research: argumentation-based approaches to LLM reasoning, self-correction methods for language models, and formal theories of belief change in argumentation.

**Argumentation and LLMs.** Vassiliades et al. (Vassiliades, Bassiliades, and Patkos 2021) survey argumentation for explainable AI; our work instantiates this vision with a concrete repair operator for LLM self-explanations. ArgLLMs (Freedman et al. 2025) constructs Dung-style graphs from LLM claims but treats verification as terminal, with no

update mechanism. ARGORA (Jin et al. 2026) orchestrates multi-agent argumentation-mediated dialogue with causal semantics, but its correction operates through re-deliberation rather than a formally defined repair operator. MQArgEng (Castagna et al. 2024) demonstrates that modular argumentation engines improve LLM reasoning but does not address explanation maintenance. ARGUS differs from all three by providing a minimal-change repair operator with AGM-compliant guarantees. Bengel and Thimm (Bengel and Thimm 2025) introduce *sequence explanations* tracing why arguments are accepted; ARGUS addresses the dual question of *how to restore* acceptance, and the two could be composed. We adopt Dung-style abstract argumentation rather than ASPIC<sup>+</sup> (Modgil and Prakken 2014) because the complexity bounds we exploit (Theorem 14) are established for this setting.

**Self-Correction and Revision.** Self-Refine (Madaan et al. 2023) and Reflexion (Shinn et al. 2023) iteratively rewrite LLM outputs but without formal minimality guarantees—previously correct reasoning may be silently discarded. Huang et al. (Huang et al. 2024) demonstrate that LLMs cannot self-correct without external feedback. RARR (Gao et al. 2023) retrieves evidence for revision but targets surface-level attribution; SelfCheckGPT (Manakul, Liusie, and Gales 2023) detects hallucinations but provides no repair mechanism. Chain-of-Verification (Dhuliawala et al. 2024) and CRITIC (Gou et al. 2024) improve factual accuracy but lack formal preservation guarantees. Matton et al. (Matton et al. 2025) measure faithfulness through counterfactual interventions and Bayesian causal models, demonstrating that instruction-tuned models produce more faithful explanations; our evaluation adopts a similar counterfactual methodology but applies it to argumentation-structured explanations. In contrast, ARGUS formalizes the repair search space as edits to an argumentation framework, bounds the cost of change, and guarantees that unaffected reasoning steps are preserved.

**Belief Revision and Argumentation Dynamics.** The AGM theory (Alchourrón, Gärdenfors, and Makinson 1985) and the revision/update distinction (Katsuno and Mendelzon 1992) provide the classical foundations for principled belief change. Hase et al. (Hase et al. 2024) argue that model editing in LLMs is fundamentally a belief revision problem and identify challenges in applying AGM rationality criteria to neural knowledge stores; our work sidesteps these challenges by operating on an *external* argumentation structure rather than on model parameters, making the AGM postulates directly applicable. Our evidence update  $\Delta$  is closer to the Katsuno–Mendelzon notion of *update* (adapting beliefs to a changed world) than to *revision* (incorporating new information about a static world), since each  $\Delta$  reflects genuinely new evidence rather than a correction of prior beliefs; however, we adopt the AGM postulates as rationality criteria because the minimal-change desiderata they formalize are independent of this distinction. In argumentation, Cayrol et al. (Cayrol, de Saint-Cyr, and Lagasquie-Schiex 2020) and Baumann and Brewka (Baumann and Brewka 2010) study how structural modifications affect extensions and the complexity of enforcement; Coste-Marquis et al. (Coste-



(a)  $F_0$ : Initial framework ( $a_4$  accepted)  
 (b)  $F_1$ : After evidence update ( $a_4$  rejected)  
 (c)  $F_2$ : After repair ( $a_4$  restored)

Figure 2: Evolution of the argumentation framework from Example 1. Green fill = accepted, red fill = rejected, blue dashed border = newly introduced, double border = target argument  $a_4$ . In (a),  $a_3$  defeats the differential  $a_0$ , keeping  $a_4$  accepted. In (b),  $a_5$  defeats  $a_3$ , reinstating  $a_0$  and rejecting  $a_4$ . The repair in (c) adds  $a_6$  attacking  $a_5$  to restore  $a_4$ .

Marquis et al. 2014), Wallner et al. (Wallner, Niskanen, and Järvisalo 2017), and Bisquert et al. (Bisquert et al. 2013) formalize argumentation revision as minimal status or structural change. Mailly (Mailly 2024) extends enforcement to constrained incomplete argumentation frameworks, where structural constraints limit the set of completions available for reasoning; our  $k$ -neighborhood approximation similarly constrains the search space, though we target a different problem (repair under evidence updates rather than enforcement under uncertainty). Alfano et al. (Alfano et al. 2024) develop counterfactual explanations for abstract argumentation via weak-constrained ASP encodings; their approach identifies minimal changes that would *reverse* an acceptance verdict, whereas ARGUS computes minimal changes that *restore* a verdict disrupted by external evidence. In particular, Coste-Marquis et al. enforce a desired extension through minimum structural modifications, whereas our formulation targets a single argument’s status, incorporates evidence updates as a first-class input, and supports heterogeneous cost functions that reflect argument-level confidence. Our repair operator instantiates these ideas for LLM explanation maintenance, introducing a weighted cost model tailored to argument confidence and structural role. We now formalize the core concepts underlying the ARGUS framework.

### 3 Preliminaries

#### 3.1 Abstract Argumentation Frameworks

We adopt the foundational model of (Dung 1995) as the backbone of our verification and repair pipeline.

**Definition 2** (Abstract Argumentation Framework). An abstract argumentation framework (AF) is a pair  $F = (\mathcal{A}, \mathcal{R})$  where  $\mathcal{A}$  is a finite set of arguments and  $\mathcal{R} \subseteq \mathcal{A} \times \mathcal{A}$  is a binary attack relation. We write  $a \rightsquigarrow b$  whenever  $(a, b) \in \mathcal{R}$ , meaning  $a$  attacks  $b$ .

**Example 3** (Continuing Example 1). The initial AF is  $F_0 = (\{a_0, a_1, a_2, a_3, a_4\}, \{(a_0, a_4), (a_3, a_0)\})$ , where  $a_0$  (“symptoms are non-specific”) attacks the target  $a_4$  and  $a_3$  (“Lupus commonly presents with these symptoms”) counterattacks  $a_0$ ; after the negative ANA result,  $F_1 = (\{a_0, a_1, \dots, a_5\}, \{(a_0, a_4), (a_3, a_0), (a_5, a_3)\})$ , as shown in Figure 2(a–b).

Intuitively, an argument is accepted if every objection against it can be countered; different semantics formalize this intuition with varying degrees of caution. Given an AF  $F = (\mathcal{A}, \mathcal{R})$ , a set  $S \subseteq \mathcal{A}$  is *conflict-free* if no two arguments in  $S$  attack each other. An argument  $a$  is *defended* by  $S$  if every attacker of  $a$  is attacked by some member of  $S$ . A conflict-free set  $S$  is *admissible* if it defends all its elements. The principal semantics we employ are the *grounded* extension, which is the unique minimal complete extension obtained as the least fixed point of the characteristic function; the *preferred* extensions, which are maximal admissible sets; and *stable* extensions, which are conflict-free sets that attack every argument outside themselves (Baroni et al. 2018). Throughout this paper we write  $\sigma(F)$  to denote the set of extensions of  $F$  under semantics  $\sigma \in \{gr, pr, st\}$ .

#### 3.2 Argumentation Semantics for Explanation

We now define the key notion linking argumentation semantics to explanation. An argument  $a \in \mathcal{A}$  is *credulously accepted* under  $\sigma$  if  $a$  belongs to at least one extension in  $\sigma(F)$ , and *skeptically accepted* if it belongs to every extension.

**Definition 4** (Defense Set). Given an AF  $F = (\mathcal{A}, \mathcal{R})$ , semantics  $\sigma$ , and an argument  $t \in \mathcal{A}$ , a defense set for  $t$  under  $\sigma$  is a minimal admissible set  $D \subseteq \mathcal{A}$  such that  $t \in D$  and  $D \subseteq E$  for some  $E \in \sigma(F)$ . We write  $Def_\sigma(t)$  for the collection of all such sets.

**Example 5** (Continuing Example 1). In  $F_0$  (Figure 2a),  $D = \{a_3, a_4\}$  is a defense set for  $a_4$ : it is conflict-free,  $a_3$  defends  $a_4$  by attacking  $a_0$ , and  $D$  is minimal since removing  $a_3$  would leave  $a_4$  undefended against  $a_0$ . In  $F_1$ ,  $D$  is no longer admissible because  $a_3$  is attacked by  $a_5$  with no counterattack, so the defense of  $a_4$  collapses.

When  $t$  is credulously but not skeptically accepted, defense sets exist only for the extensions containing  $t$ ; our repair targets the existence of at least one such set. Defense sets serve as formal explanations: each  $D \in Def_\sigma(t)$  identifies the smallest self-defending coalition that sustains  $t$ , transforming opaque LLM rationales into objects whose validity can be checked against argumentation semantics (Dunne and Wooldridge 2009).

### 3.3 Task Setting

We consider a setting in which an LLM receives a question  $q$  and produces an answer  $a$  with a free-form explanation  $e$ , which ARGUS transforms into a formal argumentation structure.

**Definition 6** (Explanation Verification Task). *Given a question  $q$ , an LLM-generated answer  $a$ , and an explanation  $e$ , the explanation verification task produces a tuple  $(G, v, \rho)$  where  $G = (\mathcal{A}, \mathcal{R})$  is an argument graph constructed from  $e$ ,  $v \in \{\text{accepted}, \text{rejected}, \text{undecided}\}$  is the verification verdict for the target argument  $a_t$  representing  $a$  under semantics  $\sigma$ , and  $\rho$  is an optional repair operator applied when  $v \neq \text{accepted}$ . An evidence update  $\Delta = (\mathcal{A}^+, \mathcal{R}^+, \mathcal{A}^-, \mathcal{R}^-)$  specifies new arguments and attacks to be added or removed, reflecting newly available facts or counterarguments.*

**Example 7** (Continuing Example 1). *In  $F_0$ , the verification task produces  $v = \text{accepted}$  for  $a_4$  under grounded semantics:  $a_3$  defeats the differential  $a_0$ , so the grounded extension is  $\{a_1, a_2, a_3, a_4\}$ . After incorporating the evidence update  $\Delta = (\{a_5\}, \{(a_5, a_3)\}, \emptyset, \emptyset)$ ,  $a_5$  defeats  $a_3$ , reinstating  $a_0$ , and the verdict becomes  $v = \text{rejected}$ , triggering the repair operator  $\rho$ .*

The target  $a_t$  is *accepted* under  $\sigma$  if it belongs to at least one  $\sigma$ -extension (credulous acceptance), and *rejected* if it belongs to no extension. Under grounded semantics, an argument may also be *undecided*—belonging to no extension yet not attacked by the grounded extension—and credulous and skeptical acceptance coincide.

### 3.4 Explanation Repair Problem

When an evidence update  $\Delta$  renders the explanation inconsistent, the system must revise the argument graph following the principle of minimal change (Alchourrón, Gärdenfors, and Makinson 1985).

**Definition 8** (Minimal-Change Repair Problem). *Let  $AF = (\mathcal{A}, \mathcal{R})$  be an AF,  $\sigma$  a semantics,  $a_t \in \mathcal{A}$  a target argument,  $s \in \{\text{IN}, \text{OUT}\}$  a desired status,  $\Delta$  an evidence update, and  $\kappa$  a strictly positive cost function ( $\kappa(o) > 0$  for every operation  $o$ ). A repair is a finite set of edit operations  $Ops = \{o_1, \dots, o_m\}$  where each  $o_i$  is one of  $\text{add\_arg}(a)$  for  $a \notin \mathcal{A} \cup \mathcal{A}^+$ ,  $\text{del\_arg}(a)$  for  $a \in \mathcal{A} \cup \mathcal{A}^+$  (which also removes all attacks incident to  $a$ ),  $\text{add\_att}(a, b)$ , or  $\text{del\_att}(a, b)$ . Let  $AF' = \text{apply}(AF, \Delta, Ops)$  denote the framework obtained by first incorporating  $\Delta$  and then executing  $Ops$ . A repair is valid if  $a_t$  has status  $s$  under  $\sigma$  in  $AF'$ , and an optimal repair minimizes  $\sum_{i=1}^m \kappa(o_i)$  over all valid repairs.*

**Example 9** (Continuing Example 1). *As shown in Figure 2(c), the repair  $Ops = \{\text{add\_arg}(a_6), \text{add\_att}(a_6, a_5)\}$  restores  $a_4$  at total cost 2 under uniform cost ( $\kappa \equiv 1$ ). The alternative  $Ops' = \{\text{del\_arg}(a_5)\}$  costs 1 but discards evidence; under structure-preserving cost with  $\kappa(\text{del\_}) = 2\kappa(\text{add\_})$ , both repairs cost 2, and domain preferences break the tie.*

The cost function  $\kappa$  encodes domain-specific preferences (e.g., deletions costlier than additions), connecting to enforcement in abstract argumentation (Baumann and Brewka

2010; Cayrol, de Saint-Cyr, and Lagasquie-Schiex 2020) while adding an explicit cost model for explanation maintenance.

## 4 The ARGUS Framework

We now present ARGUS, a four-stage pipeline (Figure 3) that transforms an unverifiable LLM rationale into a formally grounded, repairable explanation. Given a question  $q$ , an answer  $a$ , and a free-form rationale  $e$ , the pipeline proceeds through structured extraction (§4.1), relation discovery (§4.2), semantic verification (§4.3), and minimal-change repair (§4.4). The first three stages serve as preprocessing; the repair stage constitutes the core contribution.

### 4.1 Structured Extraction

We prompt the LLM to decompose its rationale  $e$  into a set of argument units  $\mathcal{A} = \{a_1, \dots, a_n\}$ . Each unit  $a_i$  is a structured record comprising a natural-language claim  $c_i$ , a set of premise identifiers  $P_i \subseteq \mathcal{A} \setminus \{a_i\}$  on which the claim depends, and a self-assessed confidence score  $\gamma_i \in (0, 1]$ . The prompt constrains the LLM to produce a JSON array of objects, each with fields `claim`, `premises`, and `confidence`, ensuring that every claim is atomic—that is, it asserts exactly one proposition that can be independently verified or rebutted. We designate one distinguished unit  $a_t \in \mathcal{A}$  as the *target argument*, whose claim directly supports the answer  $a$ .

### 4.2 Relation Discovery and Graph Construction

Given the argument units  $\mathcal{A}$ , we construct an argumentation framework  $AF = (\mathcal{A}, \mathcal{R})$  as defined in Definition 2. For every ordered pair  $(a_i, a_j)$  with  $i \neq j$ , we query a natural language inference (NLI) model—a neural classifier trained to determine entailment, contradiction, or neutrality between text pairs—to classify the relationship between  $c_i$  and  $c_j$ . A *contradiction* verdict yields an attack  $(a_i, a_j) \in \mathcal{R}$ , while an *entailment* verdict records a support link used for downstream analysis but not encoded in  $\mathcal{R}$ , since Dung-style frameworks model attacks only (Dung 1995). To improve recall on domain-specific rebuttals, we maintain an *attack template library*—a curated set of negation patterns, common exceptions, and defeasible-rule conflicts. Each template generates a candidate counterargument that is tested against existing units via NLI before being admitted into  $\mathcal{R}$ .

### 4.3 Semantic Verification

With the framework  $AF = (\mathcal{A}, \mathcal{R})$  in hand, we compute its extensions under a chosen semantics  $\sigma$  such as grounded or preferred semantics. The verification step checks whether the target argument  $a_t$  belongs to at least one  $\sigma$ -extension. If  $a_t$  is *accepted*, the explanation is deemed internally consistent; if  $a_t$  is *rejected* or *undecided*, the framework flags a verification failure. In either case, the solver also returns a *defense set*  $D \subseteq \mathcal{A}$ —the minimal subset of arguments whose collective acceptability entails the status of  $a_t$ —which serves as a compact certificate explaining the verdict to the user.

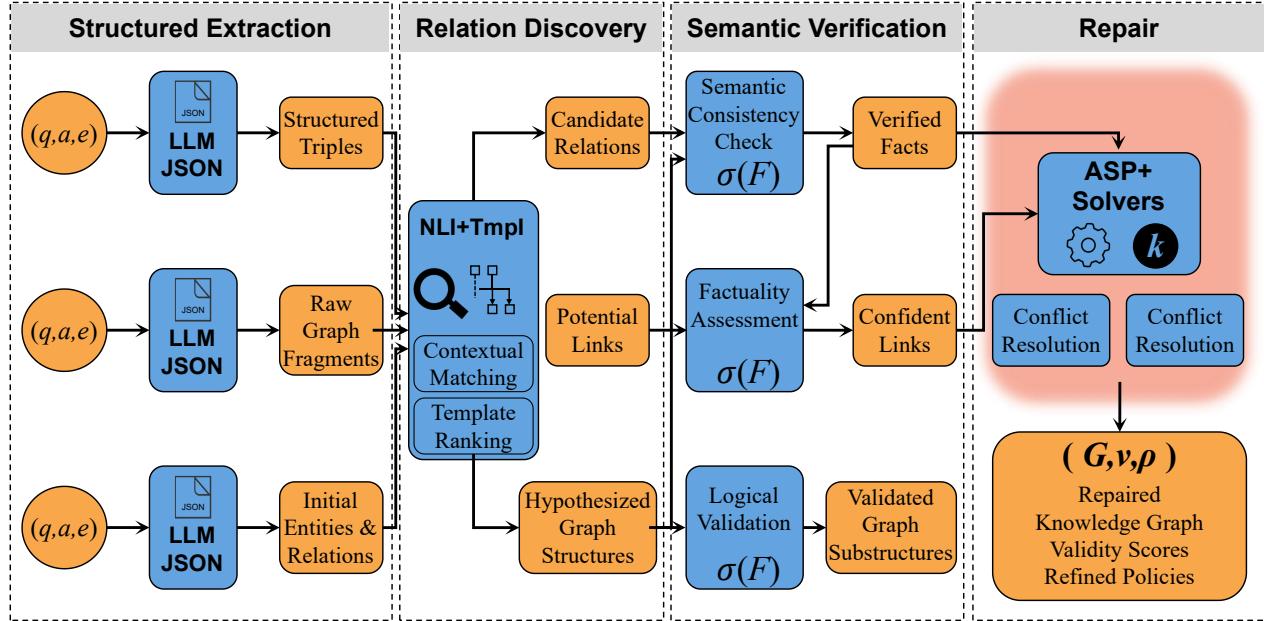


Figure 3: The ARGUS pipeline. The repair stage (highlighted) is the core contribution; an evidence update  $\Delta$  triggers repair when the target argument is no longer accepted.

#### 4.4 Minimal-Change Repair

When new evidence contradicts the current explanation or the verification step detects a failure, ARGUS repairs the argumentation framework rather than regenerating the rationale from scratch. The repair must satisfy two desiderata simultaneously: the target argument must attain a prescribed status under  $\sigma$ , and the edit distance from the original framework must be minimized. We formalize this requirement below.

**Repair Operations.** We define four elementary edit operations:  $\text{add\_arg}(a)$  and  $\text{del\_arg}(a)$  insert or remove an argument (deletions cascade to incident attacks), while  $\text{add\_att}(a_i, a_j)$  and  $\text{del\_att}(a_i, a_j)$  insert or remove attacks. A set of operations yields a repaired framework  $AF' = (\mathcal{A}', \mathcal{R}')$ .

**Cost Function.** Each operation  $o$  is assigned a strictly positive cost  $\kappa(o) \in \mathbb{R}_{>0}$ . We consider three cost models. Under *uniform cost*, every operation costs 1, so the objective reduces to minimizing the total number of edits. Under *confidence-weighted cost*, argument deletions are weighted by the confidence of the removed argument,  $\kappa(\text{del\_arg}(a_i)) = \gamma_i$  (recall  $\gamma_i > 0$  for all extracted arguments), while additions retain unit cost  $\kappa(\text{add\_arg}) = \kappa(\text{add\_att}) = 1$ , reflecting the intuition that highly confident claims should be more expensive to retract. Under *structure-preserving cost*, deletions are penalized more heavily than additions,  $\kappa(\text{del\_}\cdot) = w \cdot \kappa(\text{add\_}\cdot)$  for some  $w > 1$ , encouraging the solver to repair by augmentation rather than removal.

The repair problem is formalized in Definition 8. Given the cost function  $\kappa$  and evidence update  $\Delta$ , the solver seeks

an optimal repair—a set of edit operations of minimum total cost such that  $a_t$  attains the desired status under  $\sigma$ .

**Example 10** (Continuing Example 1). *Under confidence-weighted cost with  $\gamma_5 = 0.90$  (a verified lab result) and  $\gamma_3 = 0.75$  (a symptomatic inference), deleting  $a_5$  costs  $\kappa(\text{del\_arg}(a_5)) = 0.90$ . The augmentation repair  $\{\text{add\_arg}(a_6), \text{add\_att}(a_6, a_5)\}$  avoids removing any high-confidence argument, yielding total cost  $2\kappa(\text{add\_}\cdot)$ ; this repair is cheaper whenever  $\kappa(\text{add\_}\cdot) < 0.45$ . Under structure-preserving cost with  $w = 2$ , deleting  $a_5$  costs 2 while the augmentation still costs 2, making the two equally expensive and allowing domain preferences to break the tie.*

**ASP Encoding.** We encode the repair problem as an answer set program following the methodology of Egly et al. (Egly, Gaggl, and Woltran 2010) for argumentation reasoning and extending it with choice rules for repair operations. The encoding consists of three components; at a high level, it mirrors an integer linear program where binary variables select edits, constraints enforce semantics, and the objective minimizes cost. First, *generate rules* introduce choice atoms for each candidate operation: the solver may optionally add or delete any argument or attack within a bounded edit budget. Second, *semantics constraints* enforce that the repaired framework satisfies  $\sigma$ ; for grounded semantics, these take the form of integrity constraints requiring that every argument in the grounded extension defends itself against all attackers. Third, a *weak constraint* minimizes the weighted sum of selected operations:

$$\# \text{minimize} \{ \kappa(o) : \text{selected}(o) \}.$$

Continuing with Example 1, the choice atoms include

---

**Algorithm 1** ARGUS Repair

---

**Require:**  $AF = (\mathcal{A}, \mathcal{R})$ , semantics  $\sigma$ , target  $a_t$ , desired status  $s$ , evidence  $\Delta$ , cost function  $\kappa$ , neighborhood bound  $k$

**Ensure:** Optimal repair  $Ops^*$

- 1:  $\mathcal{A}_\Delta, \mathcal{R}_\Delta \leftarrow \text{INCORPORATE}(AF, \Delta)$
- 2:  $\mathcal{N} \leftarrow k\text{-neighborhood of } a_t \text{ in } (\mathcal{A} \cup \mathcal{A}_\Delta, \mathcal{R} \cup \mathcal{R}_\Delta)$
- 3:  $\Pi \leftarrow \text{ENCODEASP}(\mathcal{N}, \sigma, a_t, s, \kappa)$
- 4:  $M^* \leftarrow \text{SOLVE}(\Pi) \{ \text{optimal answer set} \}$
- 5:  $Ops^* \leftarrow \{o \mid \text{selected}(o) \in M^*\}$
- 6: **return**  $Ops^*$

---

455 requires that no edits are made when the current state  
456 already satisfies the goal.

457 **Theorem 11** (Adapted AGM Compliance). *Let  $AF =$   
458  $(\mathcal{A}, \mathcal{R})$  be an argumentation framework,  $\sigma$  an argumentation  
459 semantics,  $a_t$  a target argument,  $s \in \{\text{IN}, \text{OUT}\}$  a de-  
460 sired status,  $\Delta$  an evidence update, and  $\kappa$  a strictly positive  
461 cost function ( $\kappa(o) > 0$  for every operation  $o$ ). If a valid  
462 repair exists, then every optimal repair  $Ops^*$  returned by  
463 Definition 8 satisfies:*

- 464 1. **Success.** *The target  $a_t$  has status  $s$  in  $AF' =$   
465  $\text{apply}(AF, \Delta, Ops^*)$  under  $\sigma$ .*
- 466 2. **Inclusion.**  $\mathcal{A} \cap \mathcal{A}' \supseteq \mathcal{A} \setminus \{a \mid \text{del\_arg}(a) \in Ops^*\}$  and  
467  $\mathcal{R} \cap \mathcal{R}' \supseteq \mathcal{R} \setminus \{(a, b) \mid \text{del\_att}(a, b) \in Ops^*\}$ .
- 468 3. **Vacuity.** *If  $a_t$  already has status  $s$  in  $\text{apply}(AF, \Delta, \emptyset)$   
469 under  $\sigma$ , then  $Ops^* = \emptyset$  and  $\text{cost}(Ops^*) = 0$ .*

470 *Proof sketch.* Success follows directly from the validity  
471 constraint in Definition 8: any repair returned by the solver  
472 satisfies the prescribed status. Inclusion holds because ele-  
473 ments not targeted by any deletion operation are preserved  
474 by the semantics of apply; moreover, optimality ensures that  
475 every deletion in  $Ops^*$  is necessary—removing an unneces-  
476 sary  $\text{del\_arg}(a)$  would yield a valid repair of strictly lower  
477 cost ( $\kappa > 0$ ), contradicting optimality. Vacuity is immedi-  
478 ate: when no edits are needed, the empty set is valid and has  
479 cost zero, so no non-empty set can be cheaper.  $\square$

480 **Example 12** (Continuing Example 1). *Vacuity: in  $F_0$ , where  
481  $a_3$  already defeats  $a_0$  and keeps  $a_4$  accepted,  $Ops^* = \emptyset$   
482 and the repair cost is zero. Success: after incorporating  
483  $\Delta = (\{a_5\}, \{(a_5, a_3)\}, \emptyset, \emptyset)$ ,  $a_0$  reinstates and rejects  $a_4$ ;  
484 the repair  $\{\text{add\_arg}(a_6), \text{add\_att}(a_6, a_5)\}$  restores  $a_4$  to ac-  
485 cepted status by defeating  $a_5$ , which in turn restores  $a_3$  and  
486 re-defeats  $a_0$ . Inclusion: no original argument is removed—  
487 the repair only adds  $a_6$  and the attack  $(a_6, a_5)$ , preserving  
488 the entire original structure of  $F_1$ .*

489 Among the eight classical AGM postulates (Katsuno and  
490 Mendelzon 1992), *consistency* and *extensionality* also hold:  
491 consistency follows because every valid repair produces a  
492 framework with at least one  $\sigma$ -extension (under preferred  
493 semantics), and extensionality holds because the operator is  
494 defined purely over graph structure. *Recovery* fails in our  
495 setting. In Example 1, repairing  $F_1$  yields  $F_2$  by adding  $a_6$   
496 and  $(a_6, a_5)$ ; if the evidence  $a_5$  were subsequently retracted,  
497  $F_2$  would retain  $a_6$  and its attack—the original framework  
498  $F_0$  is not recovered. This asymmetry is fundamental: struc-  
499 tural additions made during repair cannot be automatically  
500 unwound by evidence retraction, unlike classical belief revi-  
501 sion where recovery ensures reversibility. *Closure*, *superex-  
502 pansion*, and *subexpansion* presuppose deductively closed  
503 belief sets—constructs without natural analogues in argu-  
504 mentation frameworks where “beliefs” are graph-structural  
505 elements rather than logical sentences. To the best of our  
506 knowledge, this is the first formal bridge between AGM ra-  
507 tionality criteria and argumentation-based explanation repair  
508 for LLM self-explanations. The contribution lies in identify-  
509 ing which AGM postulates have meaningful argumentation  
510 analogues and showing that they *characterize* the class of  
511 minimum-cost repair operators:

## 5 Theoretical Properties

441 We establish three groups of results for the ARGUS repair  
442 operator: compliance with adapted AGM postulates, com-  
443 putational complexity under the principal argumentation se-  
444 mantics, and soundness of the ASP encoding.

### 5.1 AGM Compliance

447 The AGM theory of belief revision (Alchourrón,  
448 Gärdenfors, and Makinson 1985) prescribes rationality  
449 postulates that any principled revision operator should  
450 satisfy. We adapt three core postulates—success, inclusion,  
451 and vacuity—to the argumentation repair setting. Intu-  
452 itively, success requires that the repair achieves the desired  
453 outcome; inclusion requires that the repaired framework  
454 retains as much of the original as possible; and vacuity

512 **Theorem 13** (Representation). A repair operator  $\circ$  satisfies  
 513 adapted success, inclusion, and vacuity for every AF semantics  $\sigma$ , target  $a_t$ , and evidence update  $\Delta$  if and only if there  
 514 exists a strictly positive cost function  $\kappa$  such that  $\circ$  returns a  
 515 minimum-cost valid repair under  $\kappa$ .  
 516

517 *Proof sketch.* ( $\Rightarrow$ ) Theorem 11 establishes that every  
 518 minimum-cost repair under positive  $\kappa$  satisfies all three postu-  
 519 lates. ( $\Leftarrow$ ) Given an operator satisfying the three postu-  
 520 lates, define  $\kappa(o) = 1$  for every operation  $o$ . Success guaran-  
 521 tees validity; vacuity ensures the empty set is returned when  
 522 no repair is needed, so any non-empty output incurs pos-  
 523 itive cost; inclusion ensures every operation in the output  
 524 is necessary, since removing any one would either violate  
 525 success or produce a valid repair of strictly lower cost—  
 526 contradicting the assumption that the operator already re-  
 527 turns a repair satisfying inclusion. Hence the output is a  
 528 minimum-cost valid repair under unit cost. The full con-  
 529 struction for general  $\kappa$  appears in Appendix D.  $\square$

## 5.2 Computational Complexity

531 The complexity of the repair problem depends critically  
 532 on the choice of argumentation semantics. Since the re-  
 533 pair problem reduces to enforcement after incorporating  $\Delta$ ,  
 534 it inherits the complexity landscape of extension enforce-  
 535 ment (Dunne and Wooldridge 2009; Dvořák and Dunne  
 536 2018); the additional overhead of processing  $\Delta$  and eval-  
 537 uating heterogeneous cost functions is polynomial and does  
 538 not alter the complexity class. Our results assume credulous  
 539 acceptance (as defined in §3).

540 **Theorem 14** (Repair Complexity). *The decision version*  
 541 *of the minimal-change repair problem*—“does there exist a  
 542 *valid repair of cost at most  $C$ ?*”—*has the following com-*  
 543 *plexity under credulous acceptance:*

- 544 1. Under grounded semantics, the problem is in  $P$ .
- 545 2. Under preferred and stable semantics, the problem is NP-  
 546 complete.

547 Under skeptical acceptance with stable semantics, the prob-  
 548 lem rises to  $\Sigma_2^P$ -completeness.

549 *Proof sketch.* For grounded semantics, the unique grounded  
 550 extension is computable in polynomial time via the char-  
 551 acteristic function (Dung 1995). Membership in  $P$  follows  
 552 by reduction to grounded enforcement, which Dvořák and  
 553 Dunne (Dvořák and Dunne 2018) showed is solvable in  
 554 polynomial time by exploiting the monotonicity of the char-  
 555 acteristic function. Our repair problem reduces to enforce-  
 556 ment: we first incorporate the evidence update  $\Delta$  into the  
 557 framework and then seek a minimum-cost set of edit opera-  
 558 tions that enforces the target argument’s desired acceptabil-  
 559 ity status; since both the incorporation of  $\Delta$  and the verifi-  
 560 cation of any candidate repair via the grounded extension  
 561 are polynomial, the overall decision problem is in  $P$ . For  
 562 preferred semantics, hardness reduces from NP-hard exten-  
 563 sion enforcement (Baumann and Brewka 2010); member-  
 564 ship in NP follows since a valid repair paired with a wit-  
 565 nessing admissible set containing  $a_t$  can be guessed and ver-  
 566 ified in polynomial time. For stable semantics under cred-  
 567 ulous acceptance, membership in NP follows by the same

568 certificate argument: a repair paired with a witnessing sta-  
 569 ble extension is verifiable in polynomial time. Under skep-  
 570 tical acceptance, verifying that *every* stable extension ac-  
 571 cepts the target is co-NP-hard (Dvořák and Dunne 2018),  
 572 yielding  $\Sigma_2^P$ -completeness. Full reductions follow standard  
 573 techniques from the enforcement literature (Baumann and  
 574 Brewka 2010; Wallner, Niskanen, and Järvisalo 2017).  $\square$

575 Note that the reduction to enforcement establishes com-  
 576 plexity bounds but does not subsume the repair problem,  
 577 which additionally involves evidence updates  $\Delta$ , heteroge-  
 578 neous cost functions, and NLI-grounded candidate genera-  
 579 tion (§4.2). These results motivate the  $k$ -neighborhood ap-  
 580 proximation (§4.4), ensuring tractability under preferred se-  
 581 mantics for practical framework sizes.

## 5.3 Soundness of the ASP Encoding

582 **Proposition 15** (Encoding Correctness). *The ASP encoding*  
 583 *described in §4.4, when applied to the full framework with-*  
 584 *out  $k$ -neighborhood restriction, is sound and complete with*  
 585 *respect to optimal repairs under grounded and preferred se-*  
 586 *mannets. That is, every optimal answer set of the program*  
 587 *corresponds to a valid minimum-cost repair, and every valid*  
 588 *minimum-cost repair has a corresponding optimal answer*  
 589 *set.*

591 The proof follows from the established correctness of  
 592 the argumentation encodings of Egly et al. (Egly, Gaggl,  
 593 and Woltran 2010) composed with the standard semantics  
 594 of weak constraints in ASP solvers such as *clingo* (Gebser  
 595 et al. 2019). The composition is sound because the gen-  
 596 erate rules enumerate exactly the feasible edit operations  
 597 and the integrity constraints enforce the semantics of the re-  
 598 paired framework, while the optimization directive selects  
 599 minimum-cost solutions. We next evaluate whether these  
 600 theoretical properties hold in practice and measure the em-  
 601 pirical gains of the ARGUS repair operator.

## 6 Experimental Evaluation

602 We evaluate ARGUS on two established benchmarks to an-  
 603 swer three questions: (Q1) Do the formal properties from §5  
 604 hold in practice? (Q2) Does the minimal-change repair op-  
 605 erator improve faithfulness and contestability w.r.t. existing  
 606 baselines? (Q3) What is the empirical cost of repair?

607 We evaluate on 500 randomly sampled instances (seed 42)  
 608 from HotpotQA (Yang et al. 2018) (multi-hop QA) and 500  
 609 from FEVER (Thorne et al. 2018) (fact verification). For  
 610 each instance, we withhold one gold supporting fact dur-  
 611 ing explanation generation and reintroduce it as an evidence  
 612 update  $\Delta$ , producing adversarial updates that target the rea-  
 613 soning chain. GPT-4o (gpt-4o-2024-11-20) (OpenAI  
 614 2023) generates initial explanations (temperature 0.2); rela-  
 615 tion discovery uses DeBERTa-v3-large (MultiNLI, thresh-  
 616 old 0.7); repairs are computed by clingo 5.6 with  $k=3$  under  
 617 uniform cost. Results are averaged over 5 runs ( $\text{std} \leq 0.02$   
 618 for accuracy,  $\leq 0.4$  for cost). Further details on the with-  
 619 holding methodology, hardware, and reproducibility are pro-  
 620 vided in Appendix E.

621 We measure six metrics. *Faithfulness* is the fraction  
 622 of argument units whose removal (counterfactual ablation)

changes the answer; baselines without structured units undergo the same LLM-based decomposition before ablation (details in Appendix E). *Contestability* is the fraction of gold counterarguments correctly integrated as attacks; gold counterarguments are derived from the withheld supporting facts (Appendix E). *Repair accuracy* records answer correctness after repair, and *repair cost* counts edit operations per Definition 8. *Coherence* measures the semantic consistency of repaired explanations via BERTScore (Zhang et al. 2020) between repaired and original explanations (methods without repair receive N/A). *Solve time* is the wall-clock time per instance including all repair computation.

We compare against nine baselines spanning three categories. *Self-correction methods*: SelfCheckGPT (Manakul, Liusie, and Gales 2023), Self-Refine (Madaan et al. 2023), Reflexion (Shinn et al. 2023), and RARR (Gao et al. 2023). *Verification-oriented methods*: CoT-Verifier (Ling et al. 2023), ArgLLMs (Freedman et al. 2025), FLARE (Jiang et al. 2023), and FactScore (Min et al. 2023); ArgLLMs, CoT-Verifier, and FactScore lack repair (marked N/A). *Argumentation-based*: ARGORA (Jin et al. 2026) and a naïve Regenerate baseline that re-prompts with updated evidence. For self-correction baselines, repair cost counts regenerated argument units (up to 3 rounds); cost measures are not directly commensurable with ARGUS’s structural graph edits (see Appendix E). Chain-of-Verification (Dhuliawala et al. 2024) and CRITIC (Gou et al. 2024) are excluded as they operate at generation time rather than post-hoc repair.

Table 1 summarizes the main results. ARGUS achieves the highest faithfulness (0.847/0.829) and contestability (0.791/0.768), with relative improvements of 10.3% in faithfulness and 14.5% in contestability over ARGORA (all  $p < 0.001$ , Bonferroni-corrected  $z$ -test). Among repair-capable methods, ARGUS requires the fewest operations (3.2 vs. 5.1 for ARGORA), validating the minimal-change objective. The two new retrieval-augmented baselines—FLARE and FactScore—achieve moderate faithfulness through improved evidence grounding but lack argumentation structure, resulting in low contestability (.505/.482 and .558/.535). FLARE’s repair cost is the highest (8.8/8.2) due to iterative retrieval rounds without structural guidance. The naïve Regenerate baseline achieves the lowest faithfulness (.709/.695) and fastest time (0.5s) but cannot produce contestability or coherence scores, confirming that argumentation structure is essential for principled repair.

ARGUS also achieves the highest coherence (.82/.80), indicating that minimal-change repair preserves explanation structure better than rewriting-based methods like Self-Refine (.72/.70). The solve time (0.55s/0.47s) is competitive with verification-only methods and 3–10× faster than self-correction methods that require multiple LLM rounds, reflecting the efficiency of ASP-based repair over iterative LLM regeneration. The formal properties from §5 are confirmed empirically: success and inclusion hold by construction of the ASP encoding; vacuity holds without exception, covering 5% of HotpotQA and 8% of FEVER instances where the withheld fact was not on the reasoning path. Grounded-semantics solve times averaged 0.12s and preferred 0.43s per instance.

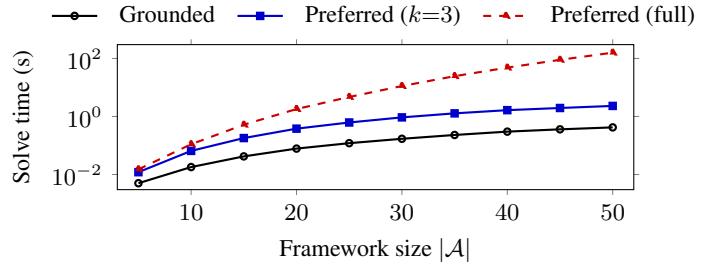


Figure 4: Scalability of ARGUS repair under grounded,  $k$ -neighborhood preferred ( $k=3$ ), and unconstrained preferred semantics. The log-scale y-axis confirms polynomial scaling for grounded repair (Theorem 14) and the effectiveness of the  $k$ -neighborhood approximation.

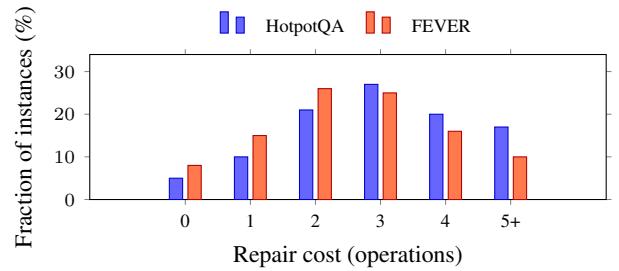


Figure 5: Distribution of repair costs. 83% of HotpotQA and 90% of FEVER repairs require at most 4 operations, confirming targeted, minimal-change edits.

Figure 4 traces solve time on synthetic Erdős–Rényi frameworks (attack probability 0.15, 50 instances per size), confirming polynomial scaling for grounded semantics (Theorem 14). The  $k$ -neighborhood approximation keeps preferred repair tractable up to  $|A|=50$  (2.31s vs. 158.4s unconstrained). Since frameworks in our experiments average 6.8 arguments, both semantics are well within practical time budgets.

Table 2 reports ablation results on both benchmarks; trends are consistent across datasets. Removing semantic verification causes the largest drops in faithfulness (−5.4pp/−5.4pp) and contestability (−7.7pp/−7.6pp), confirming that formal verification is the most critical component. Replacing minimal-change with unconstrained repair preserves faithfulness but increases cost from 3.2/2.8 to 5.7/5.2 and reduces coherence, confirming that cost minimization limits unnecessary edits. Removing attack templates reduces contestability by 9.3pp/9.0pp while leaving faithfulness largely intact. Grounded-only semantics yields the fastest solve time (0.15s/0.12s) with modest metric decreases; the gap is small because 97% of frameworks have a single preferred extension coinciding with the grounded one. Sensitivity analysis and a qualitative repair example appear in Appendices B–A.

Figure 5 shows repair cost distributions concentrated at low cost (means 3.2/2.8 operations), confirming that most evidence updates require only local adjustments. The modal cost is 3 on HotpotQA (27%) and 2 on FEVER (26%), re-

Table 1: Main results on HotpotQA and FEVER. Best in **bold**; runner-up underlined.  $\uparrow$  = higher is better,  $\downarrow$  = lower is better. N/A = method lacks repair or coherence functionality.  $^{\dagger}$ Naïve re-prompting baseline (destroys argumentation structure).

Method	HotpotQA							FEVER						
	Faith $\uparrow$	Cont $\uparrow$	RAcc $\uparrow$	RCost $\downarrow$	Coher $\uparrow$	Time $\downarrow$	Faith $\uparrow$	Cont $\uparrow$	RAcc $\uparrow$	RCost $\downarrow$	Coher $\uparrow$	Time $\downarrow$		
<i>Self-Correction Methods</i>														
SelfCheckGPT	.693	.524	.701	8.4	.68	2.8	.674	.498	.685	7.9	.66	2.5		
Self-Refine	.712	.541	.736	7.1	.72	4.5	.698	.519	.721	6.8	.70	4.2		
Reflexion	.724	.563	.752	6.6	.73	5.8	.709	.537	.738	6.2	.71	5.3		
RARR	.738	.547	.769	5.8	.71	3.2	.721	.531	.754	5.5	.69	2.9		
<i>Verification-Oriented (incl. Retrieval-Augmented)</i>														
CoT-Verifier	.751	.589	N/A	N/A	N/A	1.5	.733	.561	N/A	N/A	N/A	1.3		
ArgLLMs	<u>.754</u>	<u>.667</u>	N/A	N/A	N/A	2.1	<u>.741</u>	<u>.649</u>	N/A	N/A	N/A	1.8		
FLARE	.715	.505	.728	8.8	.74	3.8	.698	.482	.712	8.2	.72	3.5		
FactScore	.742	.558	N/A	N/A	N/A	2.5	.728	.535	N/A	N/A	N/A	2.2		
<i>Argumentation-Based</i>														
ARGORA	.768	.691	<u>.801</u>	<u>5.1</u>	.75	1.8	.752	.672	<u>.788</u>	<u>4.7</u>	.73	1.5		
Regenerate $^{\dagger}$	.709	—	.743	—	.65	<b>0.5</b>	.695	—	.729	—	.63	<b>0.4</b>		
ARGUS (Ours)	<b>0.847</b>	<b>0.791</b>	<b>0.883</b>	<b>3.2</b>	<b>.82</b>	<u>0.55</u>	<b>0.829</b>	<b>0.768</b>	<b>0.871</b>	<b>2.8</b>	<b>.80</b>	<u>0.47</u>		

Table 2: Ablation study on HotpotQA and FEVER. Each row removes one component from the full ARGUS pipeline. Best in **bold**.

Variant	HotpotQA							FEVER						
	Faith $\uparrow$	Cont $\uparrow$	RAcc $\uparrow$	RCost $\downarrow$	Coher $\uparrow$	Time $\downarrow$	Faith $\uparrow$	Cont $\uparrow$	RAcc $\uparrow$	RCost $\downarrow$	Coher $\uparrow$	Time $\downarrow$		
Full ARGUS	<b>0.847</b>	<b>0.791</b>	<b>0.883</b>	<b>3.2</b>	<b>.82</b>	.55	<b>0.829</b>	<b>0.768</b>	<b>0.871</b>	<b>2.8</b>	<b>.80</b>	.47		
w/o Semantic Verif.	.793	.714	.832	4.1	.76	.52	.775	.692	.818	3.8	.74	.44		
w/o Minimal-Change	.841	.783	.856	5.7	.78	.58	.823	.761	.842	5.2	.76	.49		
w/o Attack Templ.	.821	.698	.859	3.5	.80	.53	.804	.678	.845	3.2	.78	.45		
Grounded Only	.839	.772	.871	<b>3.0</b>	.81	<b>.15</b>	.822	.752	.858	<b>2.6</b>	.79	<b>.12</b>		

flecting FEVER’s simpler reasoning chains.

A pilot human evaluation (Appendix F) corroborates the automatic metrics: two annotators rated 75 HotpotQA instances in a blind design, preferring ARGUS in 68% of comparisons vs. Self-Refine in 19% ( $\kappa=0.62$ ). ARGUS received higher faithfulness (3.9 vs. 3.4 on a 5-point Likert scale,  $p < 0.001$ ) and coherence ratings (4.1 vs. 3.8,  $p = 0.012$ ). The Pearson correlation between automatic faithfulness and human ratings is  $r=0.78$  ( $p<0.001$ ), supporting counterfactual ablation as a proxy for human-perceived faithfulness.

## 7 Conclusion

We presented ARGUS, a framework that structures LLM self-explanations as argumentation frameworks, verifies them against formal semantics, and repairs them at minimum cost when new evidence arrives. The minimal-change repair operator satisfies adapted AGM postulates—success, inclusion, and vacuity—and a representation theorem shows that these three postulates *characterize* the class of minimum-cost repair operators under positive costs, providing formal guarantees absent from existing approaches. We established that the repair problem is tractable under grounded semantics and NP-complete under preferred and stable semantics, and introduced a  $k$ -neighborhood approximation that maintains scalability in practice. Experiments on HotpotQA and FEVER yielded relative improvements

of up to 10.3% in faithfulness and 14.5% in contestability over the strongest argumentation baseline, while achieving the lowest repair cost among all repair-capable methods.

Several limitations warrant discussion. First, the quality of the argumentation framework depends on the LLM’s ability to decompose rationales into atomic argument units; extraction errors propagate through the entire pipeline, and the faithfulness metric itself relies on the LLM’s consistency under ablation, providing a causal proxy rather than a ground-truth measure. Relatedly, the `add_arg` operation uses the same LLM to generate repair candidates, though the NLI and ASP verification stages serve as external checks that break self-referential bias; integrating retrieval-augmented verification for generated arguments is a natural extension. Second, while the  $k$ -neighborhood approximation handles the framework sizes encountered in our experiments, frameworks with hundreds of densely connected arguments may require more aggressive approximation strategies. Third, while a pilot human evaluation (Appendix F) confirms that automatic metrics correlate with human judgments, the evaluation relies primarily on automatic metrics over fact-checking and multi-hop QA datasets with synthetic evidence updates; a larger-scale human study with naturally occurring updates would further strengthen the empirical validation. Extending the approach to open-ended generation—where correctness is less well-defined and the target argu-

ment may lack a ground-truth referent—would require alternative acceptance criteria such as coherence-based semantics. Finally, while the framework supports multiple cost models, our experiments use uniform cost for simplicity; learning domain-specific cost functions from user feedback—including external calibration of the LLM’s self-assessed confidence scores—is a promising direction for future work.

Beyond these limitations, several avenues for future work emerge. First, composing ARGUS with sequence explanations (Bengel and Thimm 2025)—which trace *why* an argument is accepted—would yield a bidirectional explanation infrastructure that both diagnoses and repairs acceptance verdicts. Second, integrating the repair operator into retrieval-augmented generation pipelines could provide continual explanation maintenance as knowledge bases evolve, particularly in high-stakes domains such as clinical decision support and legal reasoning where audit trails of explanation changes are legally mandated. Third, the representation theorem (Theorem 13) opens the door to learning cost functions from human correction patterns, enabling the repair operator to adapt to domain-specific notions of minimality.

## References

- 762      784 Alchourrón, C. E.; Gärdenfors, P.; and Makinson, D. 1985.  
763      785 On the logic of theory change: Partial meet contraction  
764      786 and revision functions. *The Journal of Symbolic Logic*  
765      787 50(2):510–530.  
766      789 Alfano, G.; Greco, S.; Parisi, F.; and Trubitsyna, I. 2024.  
767      790 Counterfactual and semifactual explanations in abstract ar-  
768      791 gumentation: Formal foundations, complexity and compu-  
769      792 tation. In *Proceedings of the 21st International Conference*  
770      793 *on Principles of Knowledge Representation and Reasoning*  
771      794 (*KR*), 22–32.  
772      795 Baroni, P.; Gabbay, D.; Giacomin, M.; and van der Torre,  
773      796 L. 2018. *Handbook of Formal Argumentation*. College  
774      797 Publications.  
775      798 Baumann, R., and Brewka, G. 2010. Expanding argumenta-  
776      799 tion frameworks: Enforcing and monotonicity results. *Com-  
777      800 putational Models of Argument* 75–86.  
778      801 Bengel, L., and Thimm, M. 2025. Sequence explanations for  
779      802 acceptance in abstract argumentation. In *Proceedings of the*  
780      803 *22nd International Conference on Principles of Knowledge*  
781      804 *Representation and Reasoning* (*KR*).  
782      805 Bisquert, P.; Cayrol, C.; de Saint-Cyr, F. D.; and Lagasquie-  
783      806 Schiex, M.-C. 2013. A change model for argumentation  
784      807 frameworks. In *International Workshop on Computational*  
785      808 *Logic in Multi-Agent Systems*, 59–76. Springer.  
786      809 Castagna, F.; Ruiz-Dolz, R.; Freedman, G.; and Hunter, A.  
787      810 2024. Can formal argumentative reasoning enhance LLMs  
788      811 performances? *arXiv preprint arXiv:2405.13036*.  
789      812 Cayrol, C.; de Saint-Cyr, F. D.; and Lagasquie-Schiex, M.-  
790      813 C. 2020. Change in abstract argumentation frameworks:  
791      814 Adding and removing arguments. *Journal of Artificial Intel-  
792      815 ligence Research* 68:663–707.  
793      816 Coste-Marquis, S.; Konieczny, S.; Mailly, J.-G.; and Mar-  
794      817 quis, P. 2014. On the revision of argumentation systems:  
795      818 Minimal change of arguments statuses. In *Proceedings of*  
796      819 *the 14th International Conference on Principles of Knowl-*  
797      820 *edge Representation and Reasoning* (*KR*), 52–61.  
798      821 Dhuliawala, S.; Komeili, M.; Xu, J.; Raileanu, R.; Li, X.;  
799      822 Celikyilmaz, A.; and Weston, J. 2024. Chain-of-verification  
800      823 reduces hallucination in large language models. In *Find-  
801      824 ings of the Association for Computational Linguistics: ACL*  
802      825 2024.  
803      826 Dung, P. M. 1995. On the acceptability of arguments  
804      827 and its fundamental role in nonmonotonic reasoning, logic  
805      828 programming and n-person games. *Artificial Intelligence*  
806      829 77(2):321–357.  
807      830 Dunne, P. E., and Wooldridge, M. 2009. Complexity of  
808      831 abstract argumentation. *Argumentation in Artificial Intelli-  
809      832 gence* 85–104.  
810      833 Dvořák, W., and Dunne, P. E. 2018. Computational aspects  
811      834 of abstract argumentation. *Handbook of Formal Argumenta-  
812      835 tion* 631–688.  
813      836 Egly, U.; Gaggl, S. A.; and Woltran, S. 2010. Answer-  
814      837 set programming encodings for argumentation frameworks.  
815      838 *Argument & Computation* 1(2):147–177.

- 839 Freedman, G.; Dejl, A.; Gorur, D.; Yin, X.; Rago, A.; and  
 840 Toni, F. 2025. Argumentative large language models for ex-  
 841 plainable and contestable claim verification. In *Proceedings*  
 842 *of the AAAI Conference on Artificial Intelligence (AAAI)*.  
 843 Oral presentation.
- 844 Gao, L.; Dai, Z.; Pasupat, P.; Chen, A.; Chaganty, A. T.; Fan,  
 845 Y.; Zhao, V. Y.; Lao, N.; Lee, H.; Juan, D.-C.; et al. 2023.  
 846 RARR: Researching and revising what language models say,  
 847 using language models. In *Proceedings of the 61st Annual*  
 848 *Meeting of the Association for Computational Linguistics*.
- 849 Gebser, M.; Kaminski, R.; Kaufmann, B.; and Schaub, T.  
 850 2019. Multi-shot ASP solving with clingo. *Theory and*  
 851 *Practice of Logic Programming* 19(3):477–504.
- 852 Gou, Z.; Shao, Z.; Gong, Y.; Shen, Y.; Yang, Y.; Huang, M.;  
 853 Duan, N.; and Chen, W. 2024. CRITIC: Large language  
 854 models can self-correct with tool-interactive critiquing. In  
 855 *Proceedings of the 12th International Conference on Learn-*  
 856 *ing Representations (ICLR)*.
- 857 Hase, P.; Bansal, M.; Clark, P.; and Wiegreffe, S. 2024.  
 858 Fundamental problems with model editing: How should rational  
 859 belief revision work in LLMs? In *Advances in Neural*  
 860 *Information Processing Systems*.
- 861 Huang, J.; Chen, X.; Mishra, S.; Zheng, H. S.; Yu, A. W.;  
 862 Song, X.; and Zhou, D. 2024. Large language models cannot  
 863 self-correct reasoning yet. In *Proceedings of the 12th Inter-*  
 864 *national Conference on Learning Representations (ICLR)*.
- 865 Jiang, Z.; Xu, F. F.; Gao, L.; Sun, Z.; Liu, Q.; Dwivedi-  
 866 Yu, J.; Yang, Y.; Callan, J.; and Neubig, G. 2023. Ac-  
 867 tive retrieval augmented generation. In *Proceedings of the*  
 868 *2023 Conference on Empirical Methods in Natural Lan-*  
 869 *guage Processing (EMNLP)*, 7969–7992.
- 870 Jin, Y.; Kim, H.; Kim, K.; Lee, C.; and Shin, S. 2026. AR-  
 871 GORA: Orchestrated argumentation for causally grounded  
 872 LLM reasoning and decision making. *arXiv preprint*  
 873 *arXiv:2601.21533*.
- 874 Katsuno, H., and Mendelzon, A. O. 1992. On the difference  
 875 between updating a knowledge base and revising it. *Belief*  
 876 *Revision* 183–203.
- 877 Lanham, T.; Chen, A.; Radhakrishnan, A.; Steiner, B.; Deni-  
 878 son, C.; Hernandez, D.; Li, D.; Durmus, E.; Hubinger, E.;  
 879 Kernion, J.; et al. 2023. Measuring faithfulness in chain-of-  
 880 thought reasoning. *arXiv preprint arXiv:2307.13702*.
- 881 Ling, Z.; Fang, Y.; Li, X.; Huang, Z.; Lee, M.; Memise-  
 882 vic, R.; and Su, H. 2023. Deductive verification of chain-  
 883 of-thought reasoning. In *Advances in Neural Information*  
 884 *Processing Systems*.
- 885 Madaan, A.; Tandon, N.; Gupta, P.; Hallinan, S.; Gao, L.;  
 886 Wiegreffe, S.; Alon, U.; Dziri, N.; Prabhumoye, S.; Yang,  
 887 Y.; et al. 2023. Self-refine: Iterative refinement with self-  
 888 feedback. *Advances in Neural Information Processing Sys-*  
 889 *tems*.
- 890 Mailly, J.-G. 2024. Constrained incomplete argumentation  
 891 frameworks: Expressiveness, complexity and enforcement.  
 892 *AI Communications* 37(3):299–322.
- 893 Manakul, P.; Liusie, A.; and Gales, M. J. F. 2023. Self-  
 894 CheckGPT: Zero-resource black-box hallucination detection
- 895 for generative large language models. In *Proceedings of*  
 896 *the 2023 Conference on Empirical Methods in Natural Lan-*  
 897 *guage Processing*, 9004–9017.
- 898 Matton, K.; Sekhon, A.; Menon, R.; Lunati, L.; and Burac-  
 899 cas, G. 2025. Walk the talk? Measuring the faithfulness of  
 900 large language model explanations. In *Proceedings of the*  
 901 *13th International Conference on Learning Representations*  
 902 *(ICLR)*.
- 903 Min, S.; Krishna, K.; Lyu, X.; Lewis, M.; Yih, W.-t.; Koh,  
 904 P. W.; Iyyer, M.; Zettlemoyer, L.; and Hajishirzi, H. 2023.  
 905 FActScore: Fine-grained atomic evaluation of factual pre-  
 906 cision in long form text generation. In *Proceedings of the*  
 907 *2023 Conference on Empirical Methods in Natural Lan-*  
 908 *guage Processing (EMNLP)*, 11193–11215.
- 909 Modgil, S., and Prakken, H. 2014. The ASPIC+ frame-  
 910 work for structured argumentation: A tutorial. *Argument &*  
 911 *Computation* 5(1):31–62.
- 912 OpenAI. 2023. GPT-4 technical report. *arXiv preprint*  
 913 *arXiv:2303.08774*.
- 914 Shinn, N.; Cassano, F.; Gopinath, A.; Narasimhan, K.; and  
 915 Yao, S. 2023. Reflexion: Language agents with verbal re-  
 916 inforcement learning. *Advances in Neural Information Pro-*  
 917 *cessing Systems*.
- 918 Thorne, J.; Vlachos, A.; Christodoulopoulos, C.; and Mittal,  
 919 A. 2018. FEVER: a large-scale dataset for fact extraction  
 920 and VERification. In *Proceedings of the 2018 Conference of*  
 921 *the North American Chapter of the Association for Compu-*  
 922 *tational Linguistics*, 809–819.
- 923 Vassiliades, A.; Bassiliades, N.; and Patkos, T. 2021. Argu-  
 924 mentation and explainable artificial intelligence: A survey.  
 925 *Knowledge-Based Systems* 171:102–123.
- 926 Wallner, J. P.; Niskanen, A.; and Järvisalo, M. 2017. Com-  
 927 plexity results and algorithms for extension enforcement in  
 928 abstract argumentation. In *Proceedings of the AAAI Confer-*  
 929 *ence on Artificial Intelligence*, 1088–1094.
- 930 Yang, Z.; Qi, P.; Zhang, S.; Bengio, Y.; Cohen, W. W.;  
 931 Salakhutdinov, R.; and Manning, C. D. 2018. HotpotQA: A  
 932 dataset for diverse, explainable multi-hop question answer-  
 933 ing. In *Proceedings of the 2018 Conference on Empirical*  
 934 *Methods in Natural Language Processing*, 2369–2380.
- 935 Ye, X., and Durrett, G. 2024. Are self-explanations from  
 936 large language models faithful? In *Findings of the Associa-*  
 937 *tion for Computational Linguistics: ACL 2024*.
- 938 Zhang, T.; Kishore, V.; Wu, F.; Weinberger, K. Q.; and  
 939 Artzi, Y. 2020. BERTScore: Evaluating text generation with  
 940 BERT. In *Proceedings of the 8th International Conference*  
 941 *on Learning Representations (ICLR)*.

## A Qualitative Repair Example

Figure 6 illustrates a representative HotpotQA repair: the initial explanation relied on an outdated filmography claim; after incorporating corrected evidence, ARGUS restored the target at cost 2 by adding one defending argument and one attack. By contrast, Self-Refine regenerated the entire explanation, altering five previously correct argument units—

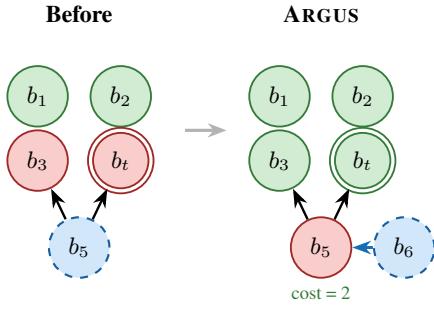


Figure 6: A HotpotQA repair example. ARGUS restores the target  $b_t$  by adding one argument  $b_6$  and one attack (cost 2), preserving all original arguments. Self-Refine regenerates 5 of 6 units.

**Construction.** Fix an enumeration of all feasible operations  $o_1, \dots, o_m$  for the given AF and  $\Delta$ . Define  $\kappa(o_i) = 1$  for all  $i$ . Let  $Ops = \circ(\mathcal{A}, \mathcal{R}, \sigma, a_t, s, \Delta)$ .

**Validity.** By success,  $a_t$  has status  $s$  in  $\text{apply}(\mathcal{A}, \mathcal{R}, \Delta, Ops)$  under  $\sigma$ , so  $Ops$  is a valid repair.

**Vacuity case.** If  $a_t$  already has status  $s$  in  $\text{apply}(\mathcal{A}, \mathcal{R}, \Delta, \emptyset)$ , then by vacuity  $Ops = \emptyset$  with cost 0, which is trivially minimum.

**Non-vacuity case.** Suppose for contradiction that there exists a valid repair  $Ops'$  with  $|Ops'| < |Ops|$ . By inclusion, every operation in  $Ops$  is necessary: for each  $o \in Ops$ , removing  $o$  would either violate success (the resulting framework does not grant  $a_t$  status  $s$ ) or leave a valid repair of cardinality  $|Ops| - 1$ , contradicting the assumption that  $\circ$  satisfies inclusion with respect to  $Ops$ .

More precisely, inclusion asserts that  $\mathcal{A} \cap \mathcal{A}' \supseteq \mathcal{A} \setminus \{a \mid \text{del\_arg}(a) \in Ops\}$ : every element not targeted by a deletion in  $Ops$  is preserved. Combined with success and  $\kappa > 0$ , this means no proper subset of  $Ops$  is both valid and of lower cost—otherwise the operator could return that subset while still satisfying all three postulates, contradicting the minimality implied by inclusion under positive costs.

Hence  $Ops$  is a minimum-cost valid repair under unit cost  $\kappa$ . For general  $\kappa$ , the same argument applies by replacing cardinality with weighted cost: inclusion ensures no operation in  $Ops$  is superfluous (removing it saves  $\kappa(o) > 0$ ), so no valid repair can have strictly lower total cost.  $\square$

## E Experimental Details

**Withholding methodology.** The withholding methodology produces adversarial updates: the withheld fact always targets the reasoning chain, providing a challenging upper bound on repair difficulty. Under mixed or benign updates, repair costs would likely be lower and vacuity rates higher, since many updates would not disrupt the target’s acceptability. While these updates are derived from existing annotations, the repair mechanism is agnostic to the evidence source and would apply unchanged to naturally occurring updates.

**Metric details.** Faithfulness is measured via counterfactual ablation: each argument unit is replaced with a semantically neutral sentence (“This claim is omitted.”) and the answer is regenerated; a unit is faithful if its removal changes the answer. For baselines that do not produce structured units, we apply the same LLM-based decomposition to their final output before computing the ablation, ensuring a uniform evaluation protocol. Gold counterarguments for contestability are derived from the withheld supporting facts by expressing each as an argument and annotating the expected attack relationships, providing a ground truth independent of the repair mechanism.

**Baseline cost commensurability.** ARGUS operations are structural graph edits (adding/removing arguments and attacks), whereas baseline costs count surface-level text replacements. Both cost measures reflect the magnitude of change to the explanation; however, the measures are not directly commensurable, so cost comparisons should be interpreted as reflecting relative parsimony within each

949 precisely the collateral damage that the minimal-change  
950 principle prevents.

## B Sensitivity Analysis

952 Pilot studies on 100 HotpotQA instances explore three de-  
953 sign choices. Confidence-weighted and structure-preserving  
954 ( $w=2$ ) cost models shift repairs toward augmentation (34–  
955 51% fewer deletions) while maintaining faithfulness and re-  
956 pair accuracy within 1 percentage point of uniform cost,  
957 confirming that the cost model affects repair *style* rather than  
958 *quality*. Varying the NLI threshold from 0.5 to 0.9 shows  
959 faithfulness is stable (0.839–0.851) while repair cost rises  
960 from 2.4 to 4.1; 0.7 balances these factors. Repair optimal-  
961 ity rises from 87.2% ( $k=1$ ) to 99.7% ( $k=3$ ) and plateaus,  
962 confirming  $k=3$  as the operating point. Using Llama-3-70B-  
963 Instruct as the extraction backbone yields faithfulness 0.813  
964 and contestability 0.762 (vs. 0.847/0.791 for GPT-4o), with  
965 comparable repair accuracy (0.867) and cost (3.4); the gap is  
966 attributable to noisier extraction rather than the repair mech-  
967 anism.

## C Error Analysis

968 Among the 0.3% of instances where minimality failed  
969 ( $k=3$ ), all involved frameworks where the only viable de-  
970 fending argument lay at distance  $\geq 4$  from the target—  
971 confirming the theoretical limitation of the  $k$ -neighborhood  
972 approximation. Repair accuracy below 1.0 arises when the  
973 LLM-generated explanation has structural errors that prop-  
974 agate through extraction: even after restoring the target ar-  
975 gument’s acceptability, the underlying answer may remain  
976 incorrect if the original decomposition was flawed.

## D Representation Theorem Proof

977 We prove the ( $\Leftarrow$ ) direction of Theorem 13 for general cost  
978 functions.

979 *Proof.* Let  $\circ$  be a repair operator satisfying adapted suc-  
980 cess, inclusion, and vacuity for every AF  $(\mathcal{A}, \mathcal{R})$ , seman-  
981 tics  $\sigma$ , target  $a_t$ , status  $s$ , and evidence update  $\Delta$ . We con-  
982 struct a strictly positive cost function  $\kappa$  such that  $\circ$  returns a  
983 minimum-cost valid repair under  $\kappa$ .

1043 paradigm. Iterative self-correction methods receive up to 3  
1044 rounds per their original protocols, while ARGUS performs  
1045 a single-pass optimal repair.

1046 **Hardware and reproducibility.** All experiments ran on  
1047 a single machine with a 20-core CPU and 64 GB RAM;  
1048 no GPU was required, as clingo runs on CPU and GPT-4o  
1049 was accessed via the OpenAI API. The complete extraction  
1050 prompt, ASP encoding, attack template library, and sampled  
1051 instance IDs will be released as an open-source toolkit upon  
1052 acceptance to facilitate reproduction.

## 1053 F Human Evaluation

1054 To validate that the automatic metrics reflect genuine qual-  
1055 ity differences perceivable by humans, we conducted a pilot  
1056 human evaluation on a random subset of 75 HotpotQA in-  
1057 stances.

1058 **Setup.** Two graduate-student annotators with NLP back-  
1059 ground independently evaluated explanation pairs produced  
1060 by ARGUS and Self-Refine (the strongest self-correction  
1061 baseline) in a blind, randomized order. For each instance,  
1062 annotators received the question, gold answer, evidence up-  
1063 date, and two candidate repaired explanations (labeled A/B  
1064 with random assignment).

1065 **Dimensions.** Annotators rated each explanation on a 5-  
1066 point Likert scale for: (1) *Faithfulness*: Does each claim in  
1067 the explanation faithfully reflect the evidence? (2) *Coher-  
1068 ence*: Is the explanation internally consistent and logically  
1069 structured? They also provided a *Preference* judgment (A  
1070 better / B better / Tie).

Table 3: Human evaluation results (75 HotpotQA instances).

Dimension	ARGUS	Self-Refine	p-value
Faithfulness (1–5)	$3.9 \pm 0.7$	$3.4 \pm 0.9$	< 0.001
Coherence (1–5)	$4.1 \pm 0.6$	$3.8 \pm 0.8$	0.012
Preference (%)	68%	19%	—
Tie (%)	13%	—	—

1071 **Agreement.** Inter-annotator agreement reached Cohen’s  
1072  $\kappa = 0.62$  (substantial) for preference and  $\kappa = 0.58$  for faith-  
1073 fulness ratings (moderate-to-substantial), confirming that  
1074 quality differences are consistently perceivable.

1075 **Correlation with automatic metrics.** The Pearson cor-  
1076 relation between average human faithfulness ratings and the  
1077 automatic faithfulness score (counterfactual ablation, §6) is  
1078  $r = 0.78$  ( $p < 0.001$ ), supporting the validity of the au-  
1079 tomatic metric as a proxy for human-perceived faithfulness.  
1080 The automatic metric tends to slightly overestimate faith-  
1081 fulness for explanations with redundant but harmless claims  
1082 (rated 4–5 by annotators but scored  $\geq 0.9$  by the metric).