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

Abstract

When large language models produce natural-language rationales, those explanations are frequently unfaithful to the model's actual reasoning—and no existing framework provides a principled way to repair them when new evidence arrives. 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 and is bidirectionally characterized by them (Representation Theorem): the decision problem is in P under grounded semantics, NP-complete under preferred and stable semantics, and Σ_2^P-complete under skeptical 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 repair-capable competing methods.

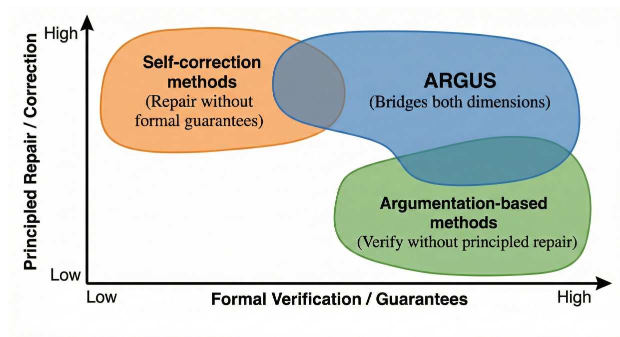


Figure 1: Qualitative positioning of ARGUS. Self-correction methods (orange) repair without formal guarantees; argumentation-based methods (green) verify without principled repair. ARGUS (blue region) bridges both dimensions.

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 Durrett, 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 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 under 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 [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 et al., 1985], which supplies the minimal-change principle, and argumentation dynamics [Cayrol et al., 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 satisfying adapted AGM postulates and bidirectionally characterized by them (Representation Theorem), with a complexity trichotomy: P under grounded, NP-complete under preferred/stable, and Σ_2^P-complete under skeptical stable semantics (§4.4–§5).
3. **(C3)** A scalable ASP encoding with a k-neighborhood approximation that preserves repair quality at 99.7% coverage (k=3) while reducing solver grounding to a tractable subproblem (§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. ten baselines (§6).

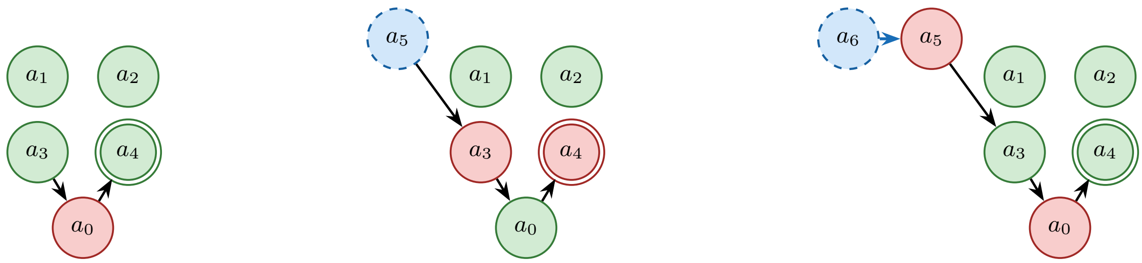
2 RELATED WORK

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

Argumentation and LLMs. Vassiliades et al. [2021] survey argumentation for explainable AI. ArgLLMs [Freedman et al., 2025] constructs Dung-style graphs from LLM claims but treats verification as terminal; ARGORA [Jin et al., 2026] orchestrates multi-agent argumentation but corrects through re-deliberation rather than formal repair; MQArgEng [Castagna et al., 2024] improves LLM reasoning with modular engines but does not address explanation maintenance. ARGUS differs by providing a minimal-change repair operator with AGM-compliant guarantees. Bengel and Thimm [2025] introduce sequence explanations tracing why arguments are accepted; ARGUS addresses the dual question of how to restore acceptance. We adopt Dung-style argumentation rather than ASPIC$^+$ [Modgil and Prakken, 2014] because the complexity bounds we exploit 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 without formal minimality guarantees; Huang et al. [2024] show that LLMs cannot self-correct without external feedback. RARR [Gao et al., 2023] targets surface-level attribution; SelfCheckGPT [Manakul et al., 2023] detects hallucinations without repair; Chain-of-Verification [Dhuliawala et al., 2024] and CRITIC [Gou et al., 2024] improve accuracy but lack preservation guarantees. Matton et al. [2025] measure faithfulness through counterfactual interventions; our evaluation applies similar probes to argumentation-structured explanations. ARGUS formalizes the repair search space, bounds change cost, and guarantees preservation of unaffected reasoning.

Belief Revision and Argumentation Dynamics. The AGM theory [Alchourrón et al., 1985] and the revision/update distinction [Katsuno and Mendelzon, 1992] provide classical foundations. Hase et al. [2024] argue that model editing is fundamentally a belief revision problem; our work sidesteps neural-level challenges by operating on an external argumentation structure. In argumentation, Cayrol et al. [2020] and Baumann and Brewka [2010] study how structural modifica-



(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.

tions affect extensions; Coste-Marquis et al. [2014], Wallner et al. [2017], and Bisquert et al. [2013] formalize argumentation revision as minimal structural change. Mailly [2024] extends enforcement to constrained incomplete frameworks; Alfano et al. [2024] develop counterfactual explanations via ASP, identifying minimal changes that reverse verdicts, whereas ARGUS restores verdicts disrupted by external evidence. ARGUS extends these foundations to LLM explanation maintenance with a weighted cost model tailored to argument confidence.

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, $F_1 = (\{a_0, a_1, \ldots, a_5\}, \{(a_0, a_4), (a_3, a_0), (a_5, a_3)\})$, 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 σ 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 σ, and an argument $t \in \mathcal{A}$, a* defense set *for t under σ 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, ρ) *where* $G = (\mathcal{A}, \mathcal{R})$ *is an argument graph constructed from e,* $v \in \{accepted, rejected, undecided\}$ *is the verification verdict for the target argument* a_t *representing a under semantics* σ*, and* ρ *is an optional repair operator applied when* $v \neq accepted$*. 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* $v = accepted$ *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* $v = rejected$*, triggering the repair operator* ρ*.*

The target a_t is *accepted* under σ if it belongs to at least one σ-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 Δ renders the explanation inconsistent, the system must revise the argument graph following the principle of minimal change [Alchourrón et al., 1985].

Definition 8 (Minimal-Change Repair Problem)**.** *Let* $AF = (\mathcal{A}, \mathcal{R})$ *be an AF,* σ *a semantics,* $a_t \in \mathcal{A}$ *a target argument,* $s \in \{\text{IN}, \text{OUT}\}$ *a desired status,* Δ *an evidence update, and* κ *a strictly positive cost function (*$\kappa(o) > 0$ *for every operation o). A* repair *is a finite set of edit operations* $Ops = \{o_1, \ldots, o_m\}$ *where each* o_i *is one of* $\mathsf{add\_arg}(a)$ *for* $a \notin \mathcal{A} \cup \mathcal{A}^+$*,* $\mathsf{del\_arg}(a)$ *for* $a \in \mathcal{A} \cup \mathcal{A}^+$ *(which also removes all attacks incident to a),* $\mathsf{add\_att}(a, b)$*, or* $\mathsf{del\_att}(a, b)$*. Let* $AF' = \mathsf{apply}(AF, \Delta, Ops)$ *denote the framework obtained by first incorporating* Δ *and then executing Ops. A repair is* valid *if* a_t *has status s under* σ *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* $Ops = \{\mathsf{add\_arg}(a_6), \mathsf{add\_att}(a_6, a_5)\}$ *restores* a_4 *at total cost* 2 *under uniform cost (*$\kappa \equiv 1$*). The alternative* $Ops' = \{\mathsf{del\_arg}(a_5)\}$ *costs* 1 *but discards evidence; under structure-preserving cost with* $\kappa(\mathsf{del\_}\cdot) = 2\kappa(\mathsf{add\_}\cdot)$*, both repairs cost* 2*, and domain preferences break the tie.*

The cost function κ encodes domain-specific preferences (e.g., deletions costlier than additions), connecting to enforcement in abstract argumentation [Baumann and Brewka, 2010, Cayrol et al., 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 $\mathcal{A} = \{a_1, \ldots, a_n\}$. 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

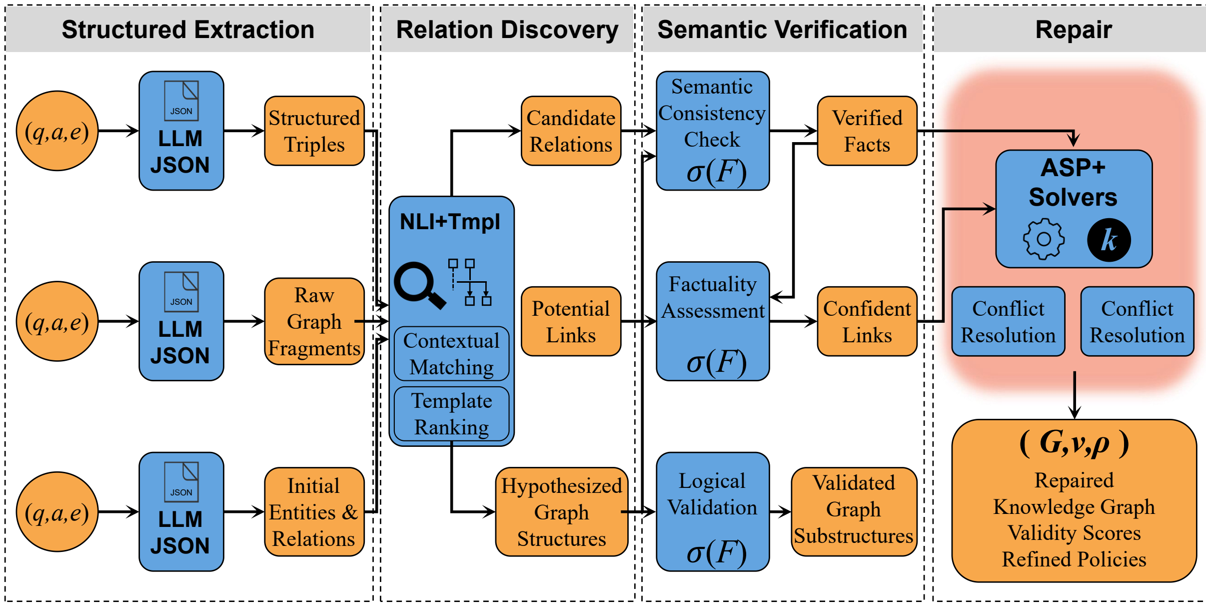


Figure 3: The ARGUS pipeline. The repair stage (highlighted) is the core contribution; an evidence update Δ triggers repair when the target argument is no longer accepted.

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 σ such as grounded or preferred semantics. The verification step checks whether the target argument a_t belongs to at least one σ-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.

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 σ, and the edit distance from the original framework must be minimized. Definition 8 formalizes this requirement.

Repair Operations. Four elementary edit operations underlie the repair: $\mathsf{add\_arg}(a)$ and $\mathsf{del\_arg}(a)$ insert or remove an argument (deletions cascade to incident attacks), while $\mathsf{add\_att}(a_i, a_j)$ and $\mathsf{del\_att}(a_i, a_j)$ 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, $\kappa(\mathsf{del\_arg}(a_i)) = \gamma_i$ (recall $\gamma_i > 0$ for all extracted arguments), while additions retain unit cost $\kappa(\mathsf{add\_arg}) = \kappa(\mathsf{add\_att}) = 1$, reflecting the intuition that highly confident claims should be more expensive to retract. Under *structure-preserving cost*, deletions are penalized more heavily than additions, $\kappa(\mathsf{del\_\cdot}) = w \cdot \kappa(\mathsf{add\_\cdot})$ 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 κ and evidence update Δ, the solver seeks an optimal repair—a set of edit operations of minimum total cost such that a_t attains the desired status under σ.

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* $\kappa(\mathsf{del\_arg}(a_5)) = 0.90$*. The augmentation repair* $\{\mathsf{add\_arg}(a_6), \mathsf{add\_att}(a_6, a_5)\}$ *avoids removing any high-confidence argument, yielding total cost* $2\kappa(\mathsf{add\_\cdot})$*; this repair is cheaper whenever* $\kappa(\mathsf{add\_\cdot}) < 0.45$*. Under structure-*

Algorithm 1 ARGUS Repair

Require: $AF = (\mathcal{A}, \mathcal{R})$, semantics σ, target a_t, desired status s, evidence Δ, cost function κ, neighborhood bound k

Ensure: Optimal repair Ops^*

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

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. [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 the neighborhood subgraph. Second, *semantics constraints* enforce that the repaired framework satisfies σ; for grounded semantics, these follow the characteristic-function fixed-point semantics of Egly et al. [2010]. Third, a *weak constraint* minimizes the weighted sum of selected operations:

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

Continuing with Example 1, the choice atoms include $\mathsf{add\_arg}(a_6)$ and $\mathsf{add\_att}(a_6, a_5)$, and the integrity constraints verify that a_4 belongs to the grounded extension of the repaired framework. Algorithm 1 summarizes the complete procedure. When the solver selects $\mathsf{add\_arg}(a)$, the natural-language claim for the new argument is generated by prompting the LLM to produce a rebuttal of the target's attacker, conditioned on the evidence update Δ; the resulting candidate is verified through the same NLI pipeline before admission.

Approximation for Scalability. Even under preferred semantics the repair problem is NP-complete (Theorem 14), rising to Σ_2^P-completeness under skeptical stable semantics [Dvořák and Dunne, 2018], so we introduce two approximation strategies. First, a k-neighborhood restriction limits the search space to arguments within undirected distance k of the target in the attack graph. The approximation is *complete* for optimal repairs whose support set lies entirely within the k-neighborhood: if the unique optimal repair modifies only arguments at distance $\leq k$ from a_t, then the neighborhood-restricted problem has the same optimum. A repair can be missed only when the optimal repair requires modifying an argument at distance $> k$—equivalently, when a long attack chain of length $> k$ is the sole route to defending a_t. In our experiments, setting k=3 recovered optimal repairs in 99.7% of cases while substantially reducing solver grounding, consistent with the shallow graph structure (median depth 3, maximum 7). Second, when ASP solvers are unavailable, beam search over repair sequences with width b provides a bounded-depth heuristic alternative. In principle, a repair valid for the subgraph may not preserve validity in the full framework if distant arguments influence the target's status; for deeper domains, k should be increased accordingly.

5 THEORETICAL PROPERTIES

We establish three groups of results for the ARGUS repair operator: compliance with adapted AGM postulates, computational complexity under the principal argumentation semantics, and soundness of the ASP encoding.

5.1 AGM COMPLIANCE

The AGM theory of belief revision [Alchourrón et al., 1985] prescribes rationality postulates that any principled revision operator should satisfy. We adapt three core postulates—success, inclusion, and vacuity—to the argumentation repair setting. Intuitively, success requires that the repair achieves the desired outcome; inclusion requires that the repaired framework retains as much of the original as possible; and vacuity requires that no edits are made when the current state already satisfies the goal.

Theorem 11 (Adapted AGM Compliance)**.** *Let $AF = (\mathcal{A}, \mathcal{R})$ be an argumentation framework, σ an argumentation semantics, a_t a target argument, $s \in \{\text{IN}, \text{OUT}\}$ a desired status, Δ an evidence update, and κ a strictly positive cost function ($\kappa(o) > 0$ for every operation o). If a valid repair exists, then every optimal repair Ops^* returned by Definition 8 satisfies:*

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

Proof sketch. Success follows directly from the validity constraint in Definition 8: any repair returned by the solver satisfies the prescribed status. Inclusion holds because elements not targeted by any deletion operation are preserved by the semantics of apply; moreover, optimality ensures that

Table 1: Main results on HotpotQA and FEVER. Best in **bold**; runner-up underlined. ↑ = higher is better, ↓ = lower is better. N/A = method lacks repair or coherence functionality. †Naïve re-prompting baseline (destroys argumentation structure).

Method	HotpotQA						FEVER					
	Faith↑	Cont↑	RAcc↑	RCost↓	Coher↑	Time↓	Faith↑	Cont↑	RAcc↑	RCost↓	Coher↑	Time↓
Self-Correction Methods												
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
Verification-Oriented (incl. Retrieval-Augmented)												
CoT-Verifier	.751	.589	N/A	N/A	N/A	1.5	.733	.561	N/A	N/A	N/A	1.3
ArgLLMs	.754	.667	N/A	N/A	N/A	2.1	.741	.649	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
Argumentation-Based												
ARGORA	.768	.691	.801	5.1	.75	1.8	.752	.672	.788	4.7	.73	1.5
Regenerate†	.709	—	.743	—	.65	**0.5**	.695	—	.729	—	.63	**0.4**
ARGUS (Ours)	**0.847**	**0.791**	**0.883**	**3.2**	**.82**	0.55	**0.829**	**0.768**	**0.871**	**2.8**	**.80**	0.47

every deletion in Ops^* is necessary—removing an unnecessary $\mathsf{del\_arg}(a)$ would yield a valid repair of strictly lower cost ($\kappa > 0$), contradicting optimality. Vacuity is immediate: when no edits are needed, the empty set is valid and has cost zero, so no non-empty set can be cheaper. □

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

Among the eight classical AGM postulates [Alchourrón et al., 1985], *consistency* and *extensionality* also hold when a valid repair exists. Consistency follows because validity requires a_t to belong to at least one σ-extension of the repaired framework. Extensionality holds because the operator is defined purely over graph structure: two evidence updates yielding identical updated frameworks produce identical repair search spaces, so the optimal repair—and the repaired framework—is the same for both. *Recovery* fails in our setting. In Example 1, repairing F_1 yields F_2 by adding a_6 and (a_6, a_5); if the evidence a_5 were subsequently retracted, F_2 would retain a_6 and its attack—the original framework F_0 is not recovered. This asymmetry is fundamental: structural additions made during repair cannot be automatically unwound by evidence retraction, unlike classical belief revision where recovery ensures reversibility. *Closure*, *superexpansion*, and *subexpansion* presuppose deductively closed belief sets—constructs without natural analogues in argumentation frameworks where "beliefs" are graph-structural elements rather than logical sentences. To the best of our knowledge, this is the first formal bridge between AGM rationality criteria and argumentation-based explanation repair for LLM self-explanations. The contribution lies in identifying which AGM postulates have meaningful argumentation analogues and showing that they *characterize* the class of minimum-cost repair operators:

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

Proof sketch. ($\Rightarrow$) Theorem 11 establishes that every minimum-cost repair under positive κ satisfies all three postulates. ($\Leftarrow$) Given an operator satisfying the three postulates, define $\kappa(o) = 1$ for every operation o. Success guarantees validity; vacuity ensures the empty set is returned when no repair is needed. We claim the returned repair Ops^* is cost-minimum. If not, some cheaper valid repair Ops' exists with $|Ops'| < |Ops^*|$. Since Ops' is a valid repair that does not contain o, the operation o is unnecessary for achieving status s—contradicting inclusion, which requires every deletion to be necessary under $\kappa > 0$. Hence $|Ops^*| \leq |Ops'|$ for all valid repairs, independent of which optimum the operator selects among ties. The full construction for general κ appears in Appendix E. □

5.2 COMPUTATIONAL COMPLEXITY

The complexity of the repair problem depends critically on the choice of argumentation semantics. Since the repair problem reduces to enforcement after incorporating Δ, it inherits the complexity landscape of extension enforcement [Dunne and Wooldridge, 2009, Dvořák and Dunne, 2018]; the additional overhead of processing Δ and evaluating heterogeneous cost functions is polynomial and does not alter the complexity class. Our results assume credulous acceptance (as defined in §3).

Theorem 14 (Repair Complexity)**.** *The decision version of the minimal-change repair problem—"does there exist a valid repair of cost at most C?"—has the following complexity under credulous acceptance:*

1. *Under grounded semantics, the problem is in **P**.*
2. *Under preferred and stable semantics, the problem is **NP**-complete.*

Under skeptical acceptance with stable semantics, the problem rises to $\mathbf{\Sigma_2^P}$-completeness.

Proof sketch. For grounded semantics, the unique grounded extension is computable in polynomial time via the characteristic function [Dung, 1995]. Membership in P follows by reduction to grounded enforcement, which Dvořák and Dunne [2018] showed is solvable in polynomial time by exploiting the monotonicity of the characteristic function. Our repair problem reduces to enforcement: we first incorporate the evidence update Δ into the framework and then seek a minimum-cost set of edit operations that enforces the target argument's desired acceptability status; since both the incorporation of Δ and the verification of any candidate repair via the grounded extension are polynomial, the overall decision problem is in P. For preferred semantics, hardness reduces from NP-hard extension enforcement [Baumann and Brewka, 2010]; membership in NP follows since a valid repair paired with a witnessing admissible set containing a_t can be guessed and verified in polynomial time. For stable semantics under credulous acceptance, NP-completeness follows: membership by the same certificate argument; hardness by the same reduction from Baumann and Brewka [2010], since every stable extension is preferred and the preferred-semantics lower bound applies. Under skeptical acceptance with stable semantics, membership in Σ_2^P follows because one can guess a repair of cost at most C and then verify—via a co-NP oracle [Dvořák and Dunne, 2018]—that every stable extension of the repaired framework contains a_t. Hardness follows by polynomial reduction from the Σ_2^P-hard problem of deciding whether argument deletion can make a target skeptically stable-accepted [Wallner et al., 2017]; since del_arg is one of our four repair operations, the lower bound carries over. Full reductions follow standard techniques from the enforcement literature [Baumann and Brewka, 2010, Wallner et al., 2017]. □

Note that the reduction to enforcement establishes complexity bounds but does not subsume the repair problem, which additionally involves evidence updates Δ, heterogeneous cost functions, and NLI-grounded candidate generation (§4.2). These results motivate the k-neighborhood approximation (§4.4), ensuring tractability under preferred semantics for practical framework sizes.

5.3 SOUNDNESS OF THE ASP ENCODING

Proposition 15 (Encoding Correctness)**.** *The ASP encoding described in §4.4, when applied to the full framework without k-neighborhood restriction, is sound and complete with respect to optimal repairs under grounded and preferred semantics. That is, every optimal answer set of the program corresponds to a valid minimum-cost repair, and every valid minimum-cost repair has a corresponding optimal answer set.*

The proof follows from the established correctness of the argumentation encodings of Egly et al. [2010] composed with the standard semantics of weak constraints in ASP solvers such as *clingo* [Gebser et al., 2019]. The composition is sound because the generate rules enumerate exactly the feasible edit operations and the integrity constraints enforce the semantics of the repaired framework, while the optimization directive selects minimum-cost solutions. We next evaluate whether these theoretical properties hold in practice and measure the empirical gains of the ARGUS repair operator.

6 EXPERIMENTAL EVALUATION

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

We sample 500 instances from HotpotQA [Yang et al., 2018], a multi-hop question-answering benchmark, and 500 from FEVER [Thorne et al., 2018], a fact-verification benchmark; instances are drawn with seed 42. For each instance, we withhold one gold supporting fact during explanation generation and reintroduce it as an evidence update Δ, producing adversarial updates that target the reasoning chain. GPT-4o [OpenAI, 2023] (`gpt-4o-2024-11-20`) generates initial explanations at temperature 0.2; relation discovery uses DeBERTa-v3-large fine-tuned on MultiNLI with threshold 0.7; repairs are computed by clingo 5.6 with k=3 under uniform cost. Results are averaged over 5 runs (std $\leq$0.02 for accuracy, $\leq$0.4 for cost); FLARE and FactScore use a single deterministic run. Further details appear in Appendix F.

Six metrics quantify performance. *Faithfulness* is the fraction of argument units whose counterfactual removal changes the answer (baselines undergo the same LLM-based decomposition; Appendix F). *Contestability* is the fraction of gold counterarguments correctly integrated as attacks; for methods without explicit argumentation frameworks, gold counterarguments are evaluated against proposition-level decompositions. *Repair accuracy* records answer correctness after repair; *repair cost* counts edit operations per Definition 8. *Coherence* measures semantic consistency via BERTScore [Zhang et al., 2020] between repaired and original explanations. *Solve time* is wall-clock time per instance.

We compare against ten baselines in three categories: self-correction methods—SelfCheckGPT [Manakul et al., 2023], Self-Refine [Madaan et al., 2023], Reflexion [Shinn et al., 2023], RARR [Gao et al., 2023]; verification-oriented methods—CoT-Verifier [Ling et al., 2023], ArgLLMs [Freedman et al., 2025], FLARE [Jiang et al., 2023], FactScore [Min et al., 2023]; and argumentation-based methods—ARGORA [Jin et al., 2026] and a naïve *Regenerate* baseline. Verification-only methods lack repair and are marked N/A. Baseline cost measures count regenerated units across up to 3 rounds, which are not directly commensurable with structural graph edits (Appendix F).

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% and 14.5% over ARGORA; all 12 pairwise differences are significant at $p < 0.001$ (Bonferroni-corrected z-tests, Cohen's $h \in [0.26, 0.38]$). Among repair-capable methods, ARGUS requires the fewest operations—3.2 vs. 5.1 for ARGORA—validating the minimal-change objective. The naïve Regenerate baseline achieves the fastest solve time (0.5 s) but its coherence (.65/.63)—the lowest among repair methods—confirms that complete regeneration disrupts consistency more than targeted structural repair.

ARGUS also achieves the highest coherence (.82/.80) and an average solve time of 0.55 s/0.47 s, which is 5–10$\times$ faster than self-correction methods. The formal properties from §5 are confirmed empirically: success and inclusion hold by construction; vacuity holds without exception. Scalability experiments on synthetic frameworks confirm polynomial scaling for grounded semantics and the effectiveness of the k-neighborhood approximation, keeping preferred repair tractable up to $|\mathcal{A}|$=50 (Figure 5, Appendix A). Ablation results (Table 2, Appendix A) confirm that semantic verification is the most critical component (-5.4pp faithfulness when removed), while 83% of repairs require at most 4 operations. A pilot human evaluation (Appendix G) corroborates these results: annotators preferred ARGUS in 68% of comparisons vs. Self-Refine in 19% (κ=0.62, r=0.78).

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 *bidirectionally characterize* the class of minimum-cost repair operators under positive costs, providing formal guarantees absent from existing approaches. Theoretically, the repair problem is tractable under grounded semantics, NP-complete under preferred and stable semantics, and Σ_2^P-complete under skeptical stable semantics; the k-neighborhood approximation maintains scalability in practice. Experiments on HotpotQA and FEVER yielded relative improvements of 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 point to future work. The framework's quality depends on the LLM's ability to decompose rationales into atomic units; extraction errors propagate through the pipeline, though the NLI and ASP verification stages mitigate this dependency. While the k-neighborhood approximation handles the framework sizes encountered in our experiments, densely connected frameworks with hundreds of arguments may require more aggressive strategies. The evaluation relies primarily on automatic metrics over fact-checking and multi-hop QA datasets; extending the approach to open-ended generation—where the target argument may lack a ground-truth referent—would require alternative acceptance criteria such as coherence-based semantics. Looking ahead, composing ARGUS with sequence explanations [Bengel and Thimm, 2025] would yield a bidirectional explanation infrastructure, and integrating the repair operator into retrieval-augmented generation pipelines could provide continual explanation maintenance as knowledge bases evolve, particularly in high-stakes domains where audit trails of explanation changes are mandated.

References

Carlos E. Alchourrón, Peter Gärdenfors, and David Makinson. On the logic of theory change: Partial meet contraction and revision functions. *The Journal of Symbolic Logic*, 50(2):510–530, 1985.

Gianvincenzo Alfano, Sergio Greco, Francesca Parisi, and Irina Trubitsyna. Counterfactual and semifactual explanations in abstract argumentation: Formal foundations, complexity and computation. In *Proceedings of the 21st International Conference on Principles of Knowledge Representation and Reasoning (KR)*, pages 22–32, 2024.

Pietro Baroni, Dov Gabbay, Massimiliano Giacomin, and Leendert van der Torre. *Handbook of Formal Argumentation*. College Publications, 2018.

Ringo Baumann and Gerhard Brewka. Expanding argumentation frameworks: Enforcing and monotonicity results. *Computational Models of Argument*, pages 75–86, 2010.

Lars Bengel and Matthias Thimm. Sequence explanations for acceptance in abstract argumentation. In *Proceedings of the 22nd International Conference on Principles of Knowledge Representation and Reasoning (KR)*, 2025.

Pierre Bisquert, Claudette Cayrol, Florence Dupin de Saint-Cyr, and Marie-Christine Lagasquie-Schiex. A change model for argumentation frameworks. In *International Workshop on Computational Logic in Multi-Agent Systems*, pages 59–76. Springer, 2013.

Federico Castagna, Ramon Ruiz-Dolz, Gabriel Freedman, and Anthony Hunter. Can formal argumentative reasoning enhance LLMs performances? *arXiv preprint arXiv:2405.13036*, 2024.

Claudette Cayrol, Florence Dupin de Saint-Cyr, and Marie-Christine Lagasquie-Schiex. Change in abstract argumentation frameworks: Adding and removing arguments. *Journal of Artificial Intelligence Research*, 68:663–707, 2020.

Sylvie Coste-Marquis, Sébastien Konieczny, Jean-Guy Mailly, and Pierre Marquis. On the revision of argumentation systems: Minimal change of arguments statuses. In *Proceedings of the 14th International Conference on Principles of Knowledge Representation and Reasoning (KR)*, pages 52–61, 2014.

Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. Chain-of-verification reduces hallucination in large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, 2024.

Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77(2):321–357, 1995.

Paul E. Dunne and Michael Wooldridge. Complexity of abstract argumentation. *Argumentation in Artificial Intelligence*, pages 85–104, 2009.

Wolfgang Dvořák and Paul E. Dunne. Computational aspects of abstract argumentation. *Handbook of Formal Argumentation*, pages 631–688, 2018.

Uwe Egly, Sarah Alice Gaggl, and Stefan Woltran. Answer-set programming encodings for argumentation frameworks. *Argument & Computation*, 1(2):147–177, 2010.

Gabriel Freedman, Adam Dejl, Deniz Gorur, Xiang Yin, Antonio Rago, and Francesca Toni. Argumentative large language models for explainable and contestable claim verification. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2025. Oral presentation.

Luyu Gao, Zhuyun Dai, Panupong Pasupat, Arun Chen, Arun Tejasvi Chaganty, Yicheng Fan, Vincent Y. Zhao, Ni Lao, Hongrae Lee, Da-Cheng Juan, et al. RARR: Researching and revising what language models say, using language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, 2023.

Martin Gebser, Roland Kaminski, Benjamin Kaufmann, and Torsten Schaub. Multi-shot ASP solving with clingo. *Theory and Practice of Logic Programming*, 19(3):477–504, 2019.

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Minlie Huang, Nan Duan, and Weizhu Chen. CRITIC: Large language models can self-correct with tool-interactive critiquing. In *Proceedings of the 12th International Conference on Learning Representations (ICLR)*, 2024.

Peter Hase, Mohit Bansal, Peter Clark, and Sarah Wiegreffe. Fundamental problems with model editing: How should rational belief revision work in LLMs? In *Advances in Neural Information Processing Systems*, 2024.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. Large language models cannot self-correct reasoning yet. In *Proceedings of the 12th International Conference on Learning Representations (ICLR)*, 2024.

Zhengbao Jiang, Frank F. Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. Active retrieval augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7969–7992, 2023.

Youngjin Jin, Hanna Kim, Kwanwoo Kim, Chanhee Lee, and Seungwon Shin. ARGORA: Orchestrated argumentation for causally grounded LLM reasoning and decision making. *arXiv preprint arXiv:2601.21533*, 2026.

Hirofumi Katsuno and Alberto O. Mendelzon. On the difference between updating a knowledge base and revising it. *Belief Revision*, pages 183–203, 1992.

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benno Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, et al. Measuring faithfulness in chain-of-thought reasoning. *arXiv preprint arXiv:2307.13702*, 2023.

Zhan Ling, Yunhao Fang, Xuanlin Li, Zhiao Huang, Mingu Lee, Roland Memisevic, and Hao Su. Deductive verification of chain-of-thought reasoning. In *Advances in Neural Information Processing Systems*, 2023.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 2023.

Jean-Guy Mailly. Constrained incomplete argumentation frameworks: Expressiveness, complexity and enforcement. *AI Communications*, 37(3):299–322, 2024.

Potsawee Manakul, Adian Liusie, and Mark J. F. Gales. SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9004–9017, 2023.

Katie Matton, Anisha Sekhon, Rakesh Menon, Luca Lunati, and Giedrius Buracas. Walk the talk? Measuring the faithfulness of large language model explanations. In *Proceedings of the 13th International Conference on Learning Representations (ICLR)*, 2025.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 11193–11215, 2023.

Sanjay Modgil and Henry Prakken. The ASPIC+ framework for structured argumentation: A tutorial. *Argument & Computation*, 5(1):31–62, 2014.

OpenAI. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 2023.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 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*, pages 809–819, 2018.

Alexandros Vassiliades, Nick Bassiliades, and Theodore Patkos. Argumentation and explainable artificial intelligence: A survey. *Knowledge-Based Systems*, 171:102–123, 2021.

Johannes Peter Wallner, Andreas Niskanen, and Matti Järvisalo. Complexity results and algorithms for extension enforcement in abstract argumentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1088–1094, 2017.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, 2018.

Xi Ye and Greg Durrett. Are self-explanations from large language models faithful? In *Findings of the Association for Computational Linguistics: ACL 2024*, 2024.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. BERTScore: Evaluating text generation with BERT. In *Proceedings of the 8th International Conference on Learning Representations (ICLR)*, 2020.

ARGUS: Argumentation-Based Minimal-Change Repair for Verifiable LLM Self-Explanations (Supplementary Material)

A ABLATION STUDY AND COST DISTRIBUTION

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

	HotpotQA / FEVER					
Variant	Faith	Cont	RAcc	RCost	Coher	Time
Full ARGUS	**.847/.829**	**.791/.768**	**.883/.871**	**3.2/2.8**	**.82/.80**	.55/.47
w/o Sem. Verif.	.793/.775	.714/.692	.832/.818	4.1/3.8	.76/.74	.52/.44
w/o Min.-Change	.841/.823	.783/.761	.856/.842	5.7/5.2	.78/.76	.58/.49
w/o Att. Templ.	.821/.804	.698/.678	.859/.845	3.5/3.2	.80/.78	.53/.45
Grounded Only	.839/.822	.772/.752	.871/.858	**3.0/2.6**	.81/.79	**.15/.12**

Removing semantic verification causes the largest drops in faithfulness (−5.4pp) and contestability (−7.7pp), confirming it as the most critical component. Replacing minimal-change with unconstrained repair preserves faithfulness but increases cost to 5.7/5.2. Removing attack templates reduces contestability by 9.3pp while reducing faithfulness by only 2.6pp. Grounded-only semantics yields the fastest solve time at the expense of modest drops; 97% of frameworks have a single preferred extension coinciding with the grounded one.

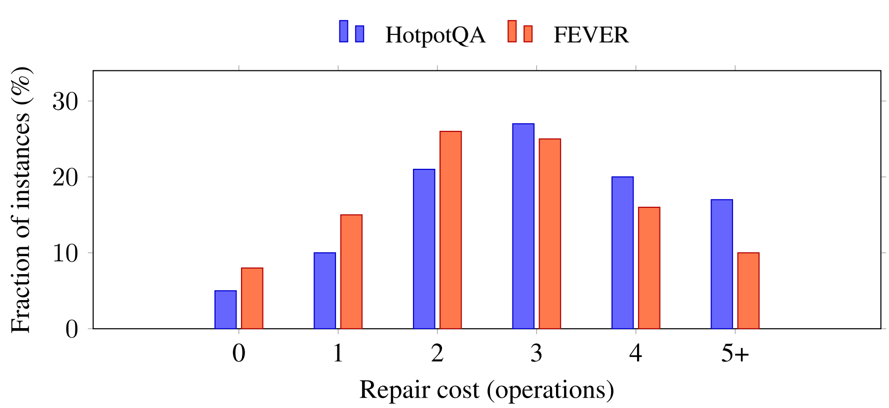


Figure 4: Distribution of repair costs. 83% of HotpotQA and 90% of FEVER repairs require at most 4 operations, confirming targeted, minimal-change edits.

Sensitivity analysis and a qualitative repair example appear in the following sections.

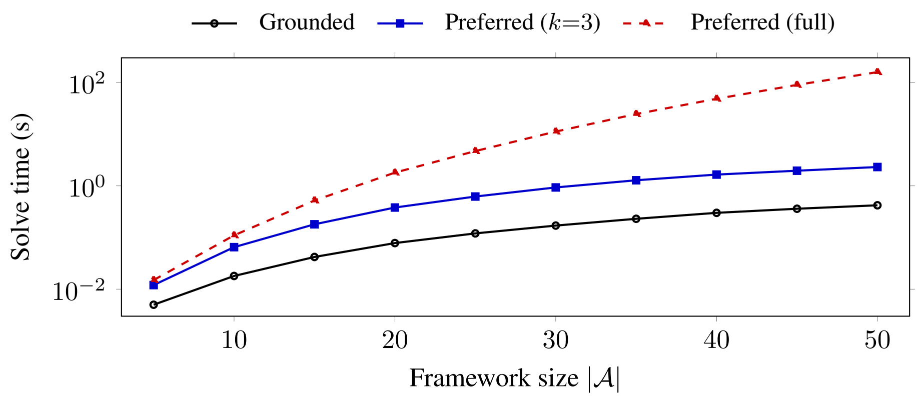


Figure 5: Scalability of ARGUS repair under grounded, k-neighborhood preferred (k=3), and unconstrained preferred semantics. Grounded repair scales polynomially (Theorem 14).

B QUALITATIVE REPAIR EXAMPLE

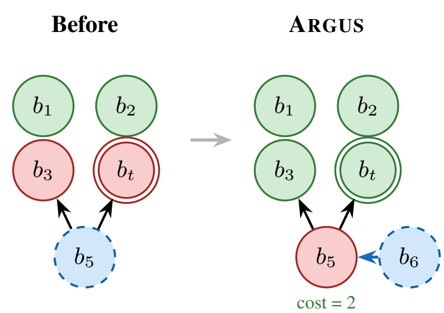


Figure 6: A HotpotQA repair example. ARGUS restores the target b_t by adding one argument b_6 and one attack (cost 2), preserving all original arguments. Self-Refine regenerates 5 of 6 units.

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—precisely the collateral damage that the minimal-change principle prevents.

C SENSITIVITY ANALYSIS

Pilot studies on 100 HotpotQA instances explore three design choices. Confidence-weighted and structure-preserving (w=2) cost models shift repairs toward augmentation (34–51% fewer deletions) while maintaining faithfulness and repair accuracy within 1 percentage point of uniform cost, confirming that the cost model affects repair *style* rather than *quality*. Varying the NLI threshold from 0.5 to 0.9 shows faithfulness is stable (0.839–0.851) while repair cost rises from 2.4 to 4.1; 0.7 balances these factors. Repair optimality rises from 87.2% (k=1) to 99.7% (k=3) and plateaus, confirming k=3 as the operating point. Using Llama-3-70B-Instruct as the extraction backbone yields faithfulness 0.813 and contestability 0.762 (vs. 0.847/0.791 for GPT-4o), with comparable repair accuracy (0.867) and cost (3.4); the gap is attributable to noisier extraction rather than the repair mechanism.

D ERROR ANALYSIS

Among the 0.3% of instances where minimality failed (k=3), all involved frameworks where the only viable defending argument lay at distance ≥ 4 from the target—confirming the theoretical limitation of the k-neighborhood approximation. Repair accuracy below 1.0 arises when the LLM-generated explanation has structural errors that propagate through extraction: even after restoring the target argument's acceptability, the underlying answer may remain incorrect if the original decomposition was flawed.

E REPRESENTATION THEOREM PROOF

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

Proof. Let $\circ$ be a repair operator satisfying adapted success, inclusion, and vacuity for every AF $(\mathcal{A}, \mathcal{R})$, semantics σ, target a_t, status s, and evidence update Δ. We construct a strictly positive cost function κ such that $\circ$ returns a minimum-cost valid repair under κ.

Construction. Fix an enumeration of all feasible operations $o_1, \ldots, o_m$ for the given AF and Δ. Define $\kappa(o_i) = 1$ for all i. Let $Ops = \circ(AF, \sigma, a_t, s, \Delta)$.

Validity. By success, a_t has status s in $\mathsf{apply}(AF, \Delta, Ops)$ under σ, so Ops is a valid repair.

Vacuity case. If a_t already has status s in $\mathsf{apply}(AF, \Delta, \emptyset)$, 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 $\mathcal{A} \cap \mathcal{A}' \supseteq \mathcal{A} \setminus \{a \mid \mathsf{del\_arg}(a) \in Ops\}$: 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 κ. For general κ, 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. $\square$

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

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

G HUMAN EVALUATION

To validate that the automatic metrics reflect genuine quality differences perceivable by humans, we conducted a pilot human evaluation on a random subset of 75 HotpotQA instances.

Setup. Two graduate-student annotators with NLP background independently evaluated explanation pairs produced by ARGUS and Self-Refine (the strongest self-correction baseline) in a blind, randomized order. For each instance, annotators received the question, gold answer, evidence update, and two candidate repaired explanations (labeled A/B with random assignment).

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

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

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

Agreement. Inter-annotator agreement reached Cohen's $\kappa = 0.62$ (substantial) for preference and $\kappa = 0.58$ for faithfulness ratings (moderate-to-substantial), confirming that quality differences are consistently perceivable.

Correlation with automatic metrics. The Pearson correlation between average human faithfulness ratings and the automatic faithfulness score (counterfactual ablation, §6) is $r = 0.78$ ($p < 0.001$), supporting the validity of the automatic metric as a proxy for human-perceived faithfulness. The automatic metric tends to slightly overestimate faithfulness for explanations with redundant but harmless claims (rated 4–5 by annotators but scored ≥ 0.9 by the metric).