Symbolic Augmentation Closes a Canonical-Equivalence Blind Spot in Neural Fact-Checkers

Anonymous ACL submission

Abstract

Large language models hallucinate numbers and units when summarizing scientific text, a failure mode that can silently invert a scientific claim. We recast the detection of such errors as *typed* verification: we introduce a five-class typed-quantity error taxonomy and a 1500-item benchmark, rewritten from PMC and arXiv sources and labeled by two independent LLM annotators with adjudication (Krippendorff's $\alpha = 0.882$). A ModernBERT encoder fine-tuned on this benchmark reaches $macro - F_1 = 0.899$, far above any off-the-shelf neural fact-checker, yet four probes expose a sharp structural blind spot: on canonical-equivalent rewrites of physically equivalent quantities (e.g., $95\,^{\circ}\mathrm{C} \leftrightarrow 368.15\,\mathrm{K}$) its accuracy collapses to 36.5%. We propose **Symbolic Augmentation**, a training-time framework that runs the modules of a symbolic verifier in reverse to generate label-preserving augmented training data. The augmentation lifts canonical-equivalence robustness to 98.2% while slightly improving in-distribution accuracy ($macro - F_1$: $0.899 \rightarrow 0.902$); the augmented encoder matches a closed-frontier LLM at no inference cost and transfers to an external benchmark (SciFact-Open binary $macro - F_1$: $0.791 \rightarrow 0.828$). Two negative results sharpen the claim: symbolic features as auxiliary encoder inputs add nothing, and symbolic silver labels scale negatively under teacher noise. Together these results identify training-time augmentation as the right integration point between symbolic and learned components.

1 Introduction

LLM-based scientific writing assistants are now widely deployed: search interfaces produce one-sentence summaries of papers (Singh et al., 2025; Asai et al., 2024), agentic pipelines compress experimental protocols (Boiko et al., 2023; M. Bran et al., 2024), and authoring tools draft method sections from raw notes (Liang et al., 2024). In all of these settings, numeric and unit hallucinations are a uniquely dangerous failure mode: a magnitude, unit, or comparator in the generated text disagrees with the source in a way that changes the scientific claim, while remaining grammatically fluent. Unlike paraphrase errors, which a careful reader can often catch from context, a single mis-stated unit can flip the safety profile of a drug candidate without any local linguistic cue (Ji et al., 2023; Zhang et al., 2023).

Consider an evidence sentence "Compound A inhibited EGFR with an IC_{50} of $12\,\mathrm{nM}$" summarized as "…with an IC_{50} of $12\,\mathrm{mg\,mL^{-1}}$." A practicing biologist sees the problem at once: nM is a molar concentration while mg/mL is a mass-volume concentration, and converting between them requires the molar mass of compound A—the two summaries support entirely different conclusions. A generic fact-check verifier, however, will at best answer "not entailed," giving no signal about *which kind* of error occurred, *which span* to repair, or whether the defect is a typo, an order-of-magnitude slip, or a category-level confusion. We argue this task should be modeled as *typed* classification, with the verifier producing not a binary verdict but one of five labels (CORRECT, UNIT ERROR, SCALE ERROR, RELATION ERROR, UNSUPPORTED) that each corresponds to a distinct symbolic axis of the quantitative claim.

To enable systematic study of this task, we construct a frozen 1500-sentence evaluation set, LLM-rewritten from real PMC (biomedical and materials science) and arXiv (computer science) sources, balanced across the five classes and dual-annotated by two independent LLMs with third-party adjudication on disagreements; Krippendorff's $\alpha = 0.882$ exceeds the standard 0.80 publishability threshold (Ford, 2004). A small probe set with a disjoint perturbation generator and a 50-item human spot-check (94% agreement) address the self-labeling circularity concern.

On this benchmark, a fine-tuned ModernBERT encoder (Warner et al., 2025; Sileo, 2024) reaches $macro - F_1 = 0.899$ (5-fold CV), far above any off-the-shelf neural fact-checker we test and within 0.01–0.07 of closed-frontier LLMs at a fraction of the cost. The encoder is, by this measure, remarkably capable—yet it exhibits a sharp *structural* blind spot. When we apply canonical-equivalent rewrites to gold CORRECT items—rewriting 95 °C as 368.15 K, or 0.5 mol/L as 500 mmol/L—its accuracy on these physically equivalent pairs drops from 96.7% to 36.5%. Three further probes (lexical synonyms, unit-name paraphrase, scientific notation) localize the failure: the encoder is robust to lexical and syntactic variation but cannot canonicalize physically equivalent numerical surfaces.

This blind spot suggests an integration question: a symbolic verifier canonicalizes units trivially via a registry, so can it supplement the encoder where the encoder is blind, and at which point in the pipeline? We answer by introducing **Symbolic Augmentation**, a training-time framework that runs the modules of a symbolic verifier in reverse to generate label-preserving training data. Four augmentation families—covering unit canonical-equivalence, comparator synonyms, unit-name paraphrase, and scientific notation—are each derived from one verifier module (§4) and tested with a per-family probe. Training with the canonical-equivalence family raises its probe accuracy from 36.5% to 98.2% while slightly improving in-distribution accuracy; the augmented encoder matches a closed-frontier LLM at no inference cost and transfers to an external benchmark. We complement these positive results with two principled negatives—symbolic features as auxiliary inputs, and the verifier as a silver-label teacher—that locate training-time augmentation as the right integration point.

Contributions.

(C1) Task and benchmark. We define a five-class typed-quantity error taxonomy with priority rules and a templated repair specification (§3), and release a 1500-item benchmark rewritten from PMC and arXiv sources, dual-annotated by two independent LLMs with third-party adjudication ($\alpha = 0.882$) and validated by a 50-item human spot-check (§5.1).

(C2) Symbolic Augmentation. We localize the fine-tuned encoder's residual failure to canonical-equivalence rewrites through a four-family probe suite, and propose a training-time framework that derives label-preserving augmentation rules from the modules of a symbolic verifier, closing the blind spot and transferring to an external benchmark (§4, §5.3).

(C3) Where symbolic information should enter. We compare five integration points and show that only training-time augmentation delivers a measurable, transferable benefit: feature-level concatenation, silver-label teaching, and inference-time ensembling all fail (§5.4).

2 Related Work

Faithfulness verification. The dominant paradigm casts faithfulness as binary or ternary entailment between a claim and its evidence. MiniCheck (Tang et al., 2024) distills a calibrated fact-checker matching GPT-4 at lower cost, and a line of recent systems extends this paradigm—Bespoke-MiniCheck-7B (Bespoke Labs, 2024), FactCG (Lei et al., 2025), VeriFastScore (Rajendhran et al., 2025), and Granite Guardian (Padhi et al., 2024)—alongside sampling- and tool-based checkers (Manakul et al., 2023; Chern et al., 2023). A parallel line moves beyond a single faithfulness scalar: ACUEval (Wan et al., 2024) types atomic units as supported, missing, or hallucinated; FineSurE (Song et al., 2024) predicts sentence-level error types; and the HalluLens benchmark (Bang et al., 2025) and Mu-SHROOM shared task (Vázquez et al., 2025) target fine-grained hallucination. These typed systems share our premise that a binary judgment is information-poor, but organize errors around *content-source* relations (extraneous, missing, unsupported) rather than the *symbolic structure* of quantitative claims; the two taxonomies are complementary. A separate line characterizes *where* fine-tuned encoders fail: probing methodology (Belinkov, 2022; Ribeiro et al., 2020; Rogers et al., 2020) and input-level blind-spot prediction (Mi et al., 2025) diagnose narrow failure modes that aggregate metrics hide. We follow this tradition in §4 with a four-family probe suite; concurrent work on OOD generalization in NLI (Delbari and Pilehvar, 2025) similarly finds that in-distribution accuracy can mask consequential robustness gaps.

Numeric and scientific claim verification. Verification targeted specifically at numerical and scientific claims is more recent. Number representation in NLP has been studied broadly (Thawani et al.,

2021; Mishra et al., 2022), but benchmarks dedicated to numerical verification have only emerged in the past two years: QuanTemp (Anand et al., 2024) and QuanTemp++ (Venktesh et al., 2025a) taxonomize claims into statistical, temporal, comparison, and interval categories, and the CheckThat! 2025 Task 3 (Venktesh et al., 2025b) adopts QuanTemp as a shared task with most submissions casting veracity as three-class NLI (Heil and Pramov, 2025; Anik et al., 2025) reaching macro-$F_1 \leq 0.57$; VerifierFC (Chungkham et al., 2025) explores test-time scaling for numerical claims. SciFact (Wadden et al., 2020), SciFact-Open (Wright et al., 2022), and SciVer (Wadden and Lo, 2021) remain the canonical scientific fact-check benchmarks, but their errors are paraphrased claims rather than unit- or magnitude-typed perturbations. Domain-adapted scientific LMs continue to appear—LLaMat (Mishra et al., 2024), HalluMat (Vangala et al., 2025), MedRAGChecker (Ji et al., 2026)—but as we show in §5, surface domain familiarity does not translate into structural reasoning over quantities. Two distinctions separate this entire line from our setting: existing taxonomies partition claims by *topic* (temporal vs. statistical), while ours partitions them by *error type* (dimension vs. scale vs. relation); and existing benchmarks do not recover *which kind* of numerical defect is present, the specific capability our taxonomy operationalizes.

Symbolic verification and data augmentation. UNITGRAPH VERIFIER sits in the lineage of symbolic and tool-augmented verification. Rule-based pipelines using quantulum-style extractors (The quantulum3 contributors, 2024) and Pint unit registries (Grecco et al., 2024) have long been used for unit canonicalization in scientific text mining but are seldom benchmarked as standalone verifiers. More recent work integrates symbolic reasoning into verification: TabVer (Aly and Vlachos, 2024) extends natural logic with set-theoretic arithmetic for tabular claims, and CoSineVerifier (Feng et al., 2025) adds symbolic execution for computation-heavy STEM answers. Concurrent work (Seo et al., 2025) similarly identifies pitfalls in fact verifiers and explores synthetic data generation as a remedy.

A parallel line uses data augmentation to inject knowledge into encoder training rather than at inference time. Counterfactual data augmentation (Zhu et al., 2023; Zhang et al., 2025) generates label-flipping rewrites of training items, typically via learned paraphrasers or LLM prompting (Whitehouse et al., 2023; Chen et al., 2023). Our Symbolic Augmentation (§4) is in this spirit but differs in that its rewrite rules are derived deterministically from the modules of a symbolic verifier—yielding stronger label guarantees—and are *label-preserving* rather than label-flipping, the appropriate shape for closing canonical-equivalence blind spots. The framework extends to any symbolic verifier whose modules expose deterministic, label-preserving rewrites.

3 Typed-Quantity Verification

3.1 Task formulation

Given an evidence sentence e and a summary s purporting to restate e's quantitative content, the verifier assigns a label y from one of five classes (CORRECT, UNIT ERROR, SCALE ERROR, RELATION ERROR, UNSUPPORTED) and optionally emits a templated repair r that, applied to s, yields a sentence consistent with e. Each non-CORRECT class corresponds to a distinct symbolic axis along which s can deviate from e—unit dimension, magnitude scale, relational structure, and unsupported-claim addition—while CORRECT covers everything else, including *surface variations that preserve the underlying physical claim*, a property central to our augmentation framework (§4). The taxonomy is dataset-agnostic; we verify transfer to an external benchmark in §5.3.

3.2 Class definitions

CORRECT. s is a paraphrase or faithful condensation of e. Crucially, this includes *canonical-equivalent rewrites*—surface forms that canonicalize to the same physical quantity through a unit registry or notational equivalence. For example, 95 °C and 368.15 K denote the same temperature, 0.5 mol/L and 500 mmol/L the same concentration, and 1.2e-9 and 0.0000000012 the same value. Such rewrites are correct restatements, not errors, even though their surface forms differ from the evidence.

UNIT ERROR. s contains a quantity whose unit has a *different physical dimension* from the evidence's. Example: "12 nM" → "12 mg mL^{-1}."

SCALE ERROR. s contains a quantity with the *same physical dimension* as the evidence's but a numerical value differing by at least one order of magnitude ($|\log_{10}(q_s/q_e)| \geq 1$). One exception applies: if two surface forms canonicalize to the

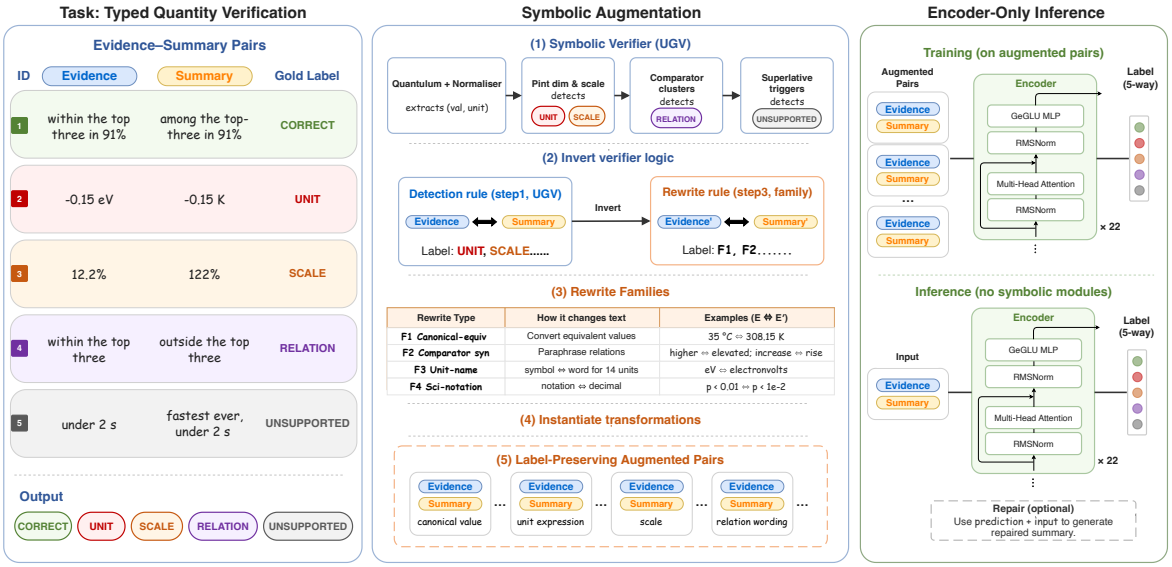

Figure 1: Overview of the typed-quantity verification task and the Symbolic Augmentation framework. **(left)** The task: each evidence–summary pair is assigned one of five typed labels (CORRECT, UNIT ERROR, SCALE ERROR, RELATION ERROR, UNSUPPORTED), with one example per label. **(center) Symbolic Augmentation**, our contribution: the symbolic verifier UGV—which extracts quantities, checks dimension and scale, and detects comparator and superlative patterns—is run in reverse. Each detection rule is inverted into a label-preserving rewrite family (F1–F4), which is instantiated to produce augmented training pairs that target the encoder's canonical-equivalence blind spot. **(right)** At inference time the fine-tuned encoder runs alone, with no symbolic modules; a templated repair is optional.

same physical quantity within 1%, the pair is a canonical-equivalent CORRECT restatement, not a magnitude error. Example: "92% conversion" → "9.2% conversion."

RELATION ERROR. s contains a *relational* discrepancy: (a) a comparator flip ("higher" → "lower"), (b) a value bound to the wrong entity, or (c) a within-order numerical change altering the claim's scope ("after 24 h" → "after 48 h").

UNSUPPORTED. s adds a claim or framing absent from e. This covers three sub-types: superlative claims ("the highest ever reported"), causal claims ("the dominant factor"), and significance claims ("statistically significant"). Numeric content may still be correct; the defect is the unsupported addition.

3.3 Priority and evaluation

Priority. A summary may exhibit multiple error types simultaneously. We resolve conflicts in descending order: UNSUPPORTED, UNIT ERROR, SCALE ERROR, RELATION ERROR, CORRECT. UNSUPPORTED dominates typed errors because it is an epistemic shift independent of the numeric content; the remaining ordering reflects axis precedence (dimension before magnitude before relation). The ordering is empirically active on 116 of 1500 items (7.7%) and is detailed further in Appendix A.

Evaluation. We evaluate verifiers along three axes: (i) classification accuracy on a frozen 1500-item test set (§5); (ii) robustness on per-family probes that apply physical-claim-preserving surface variations (§4.3); and (iii) out-of-distribution transfer to an external benchmark (§5.3). For systems that emit a templated repair, we report repair accuracy as a secondary metric (Appendix A).

4 Symbolic Augmentation

4.1 The UNITGRAPH VERIFIER

UNITGRAPH VERIFIER (UGV) is a deterministic, four-stage pipeline that mirrors the task's symbolic structure (§3.1): each stage targets one error axis from §3.2, and a fifth implicit branch returns CORRECT when no stage fires. The pipeline is CPU-only with no language model at inference time; its only non-standard dependencies are quantulum (The quantulum3 contributors, 2024) for quantity extraction and Pint (Grecco et al., 2024) for the unit registry. The full pseudocode is in Appendix B, and

Figure 1 gives an overview.

Stages 1–2: extraction and dimensional comparison. Quantulum maps each surface span to a tuple $(v, u, span)$ of value, unit, and character range. A hand-curated normalizer canonicalizes unit aliases (`M` $\rightarrow$ `mol/L`; `wt%` $\rightarrow$ percent), scientific notation, and context-implicit units. Aligned quantity pairs (q_e, q_s) are then checked for dimensional inequality ($\dim(q_s) \neq \dim(q_e)$), yielding UNIT ERROR, or for order-of-magnitude inequality, yielding SCALE ERROR—subject to the 1% canonical-equivalence override of §3.2.

Stages 3–4: relations and unsupported triggers. Stage 3 curates a small inventory of comparator clusters (e.g., the "high" family: *higher*, *elevated*, *greater*) and tags occurrences with polarity and arguments; a matched pair is inconsistent if polarities differ (flip), identities are swapped, or scope shifts. Stage 4 fires UNSUPPORTED when one of a finite set of trigger phrases (superlative, causal, significance markers; full inventory in Appendix B) appears in s but is absent from e and its surrounding context. UGV reaches $macro - F_1 = 0.537$ on the 1500-item benchmark (§5.2), exceeding every off-the-shelf neural fact-checker we test but falling well below a fine-tuned encoder.

4.2 The augmentation framework

The four UGV stages share a structural property we exploit: each is a *deterministic, label-preserving rewrite rule* over surface forms. Pint's unit registry, for instance, defines when two surface forms (95 °C and 368.15 K) denote the same physical quantity. Run in reverse, this rule generates training data: given a gold CORRECT item, rewriting its quantity to a canonical-equivalent form yields a new CORRECT item whose label is guaranteed correct, because the rule is label-preserving by construction.

We call the resulting framework **Symbolic Augmentation**. It applies four augmentation families, each derived from one UGV module:

F1: Canonical equivalence. Convert quantities to physically-equivalent units across systems (95 °C $\leftrightarrow$ 368.15 K; mg $\leftrightarrow$ g; mM $\leftrightarrow$ M; % $\leftrightarrow$ fraction).

F2: Comparator synonyms. Swap comparator words on both e and s symmetrically ("higher" $\leftrightarrow$ "elevated"; "lower" $\leftrightarrow$ "reduced").

F3: Unit-name paraphrase. Substitute unit symbols with their word forms in both e and s (`M` $\leftrightarrow$ "molar"; `K` $\leftrightarrow$ "kelvin").

F4: Scientific notation. Convert numbers between exponential and decimal forms on the summary side (`1.2e-9` $\leftrightarrow$ `0.0000000012`).

Augmentation is applied to gold CORRECT items only, so the augmented training set retains the original gold-label distribution; F1 and F4 rewrite only the summary side, while F2 and F3 rewrite both e and s symmetrically to preserve the gold relation. The framework is extensible: any symbolic verifier whose modules expose deterministic, label-preserving rewrites can be added as a family.

4.3 Per-family probes

To diagnose where the encoder fails and to measure augmentation effect, we construct a probe set for each family. A probe applies the family's rewrite to the test set's CORRECT items and asks whether the model still labels them CORRECT; probe accuracy is the fraction it classifies correctly. The four probes contain 181, 22, 30, and 25 variants for F1–F4 respectively, derived from the 333 CORRECT items in the test set. Rule definitions and probe generation share code (Appendix C), so a probe and its corresponding augmentation test exactly the same transformation. Probes thus serve a dual role: a diagnostic for where fine-tuned encoders fail (§5.2), and a benchmark for whether augmentation localizes its effect to the targeted failure (§5.3).

5 Experiments

5.1 Setup

Dataset. We evaluate on the 1500-item typed-quantity benchmark of §3. Items are LLM-rewritten from real PMC (biomedical, materials science) and arXiv (computer science) source sentences, balanced across the five classes, and dual-annotated by two independent LLM annotators with third-party adjudication (Krippendorff's $\alpha = 0.882$); a 50-item human spot-check agrees with the gold labels on 94% of items. Appendix D details the construction pipeline, including source sampling, rewrite prompting, and circularity controls.

Baselines. We compare five groups of systems. *(1) Trivial baselines:* three systems that establish the metric floor—a constant CORRECT pre-

	System	macro-F_1	Correct	Unit	Scale	Relation	Unsup.
Trivial	predict-CORRECT	0.073	0.363	0.000	0.000	0.000	0.000
	string-overlap	0.084	0.000	0.000	0.000	0.000	0.422
	unit-substring	0.225	0.391	0.437	0.000	0.000	0.299
Rule-based	quantulum + Pint	0.385	0.373	0.548	0.708	0.000	0.299
Ours (symbolic)	UGV	0.537	0.413	0.620	0.612	0.358	0.684
Off-the-shelf	VeriFastScore	0.073	0.363	0.000	0.000	0.000	0.000
	Granite Guardian	0.155	0.000	0.289	0.000	0.144	0.343
	FactCG	0.184	0.440	0.405	0.000	0.077	0.000
	Bespoke-MiniCheck-7B	0.229	0.353	0.303	0.000	0.145	0.343
	ModernBERT-NLI (zero-shot)	0.275	0.671	0.265	0.000	0.152	0.286
	MiniCheck	0.295	0.714	0.337	0.000	0.150	0.275
Fine-tuned	ModernBERT-FT (canonical)	0.899	0.855	0.923	0.896	0.892	0.930
	+ F1 Symbolic Aug.	**0.902**	0.868	0.923	0.899	0.886	0.931
	+ all-four Symbolic Aug.	0.893	0.843	0.924	0.894	0.888	0.917
Closed-frontier	DeepSeek V4-Flash	0.908	0.919	0.958	0.938	0.881	0.844
	GPT-5.5	0.951	0.957	0.975	0.958	0.940	0.923
	Claude Opus 4.7	0.973	0.983	1.000	0.967	0.959	0.959

Table 1: Main results on the 1500-item benchmark: macro-F_1 and per-class F_1 over the five error types.

dictor, a string-overlap exact-match check, and *unit-substring*, which flags a UNIT ERROR whenever a value-and-unit token in the summary is absent from the evidence. *(2) Rule-based:* quantulum (The quantulum3 contributors, 2024) paired with the Pint unit registry (Grecco et al., 2024). *(3) Our symbolic verifier:* UGV (§4.1). *(4) Off-the-shelf neural fact-checkers:* MiniCheck (Tang et al., 2024), Bespoke-MiniCheck-7B (Bespoke Labs, 2024), FactCG (Lei et al., 2025), a zero-shot ModernBERT-NLI (Sileo, 2024), Granite Guardian (Padhi et al., 2024), and VeriFastScore (Rajendhran et al., 2025). *(5) Closed-frontier LLMs:* Claude Opus 4.7, GPT-5.5, and DeepSeek V4, evaluated zero-shot. Versions, checkpoints, and invocation settings for every system are in Appendix E.

Metrics. Our primary metric is the macro-averaged F_1 score over the five classes. For the augmentation experiments we additionally report *probe accuracy*: the fraction of a family's probe items (§4.3) that the model still classifies correctly. For the out-of-distribution experiment we report binary macro-F_1 on SciFact-Open. Exact metric definitions are given in Appendix F.

Fine-tuning and augmentation. Our fine-tuned model is ModernBERT-NLI (Warner et al., 2025; Sileo, 2024) fine-tuned on the typed-quantity task and evaluated with 5-fold cross-validation. Symbolic Augmentation is applied to the gold CORRECT training items of each fold as described in §4.2. Training hyperparameters, hardware, and augmentation volumes are reported in Appendix G.

5.2 The encoder's blind spot

Table 1 reports main results, and the picture is sharp. No off-the-shelf neural system exceeds $macro - F_1 = 0.295$, and the per-class columns show why: *every off-the-shelf neural system scores exactly* 0.000 *on* SCALE ERROR. Their CORRECT scores vary widely (0.00–0.71)—some detect that a quantity is being restated, but none detect that a magnitude is wrong. The rule-based quantulum + Pint baseline (0.385) and UGV (0.537) both outperform every neural system, because explicit dimensional arithmetic is exactly what SCALE ERROR and UNIT ERROR require.

Fine-tuning changes the picture entirely: ModernBERT fine-tuned on the typed-quantity task reaches $macro - F_1 = 0.899$ (5-fold CV), within 0.07 of the strongest closed-frontier LLM (Claude Opus 4.7, 0.973) and ahead of DeepSeek V4-Flash. A modest encoder, given a few hundred labeled examples per class, is competitive with frontier systems costing orders of magnitude more per query. Yet one per-class score stands out: CORRECT is the encoder's *weakest* class at $F_1 = 0.855$, well below its UNIT ERROR (0.923) and UNSUPPORTED (0.930) scores. A model this strong should not find CORRECT hardest—unless some CORRECT items share a structure it cannot handle.

The four probes of §4.3 locate that structure. The *none* row of Table 3 reports the canonical encoder's probe accuracies: on three of four—comparator

Model	macro-F_1	F1 probe
ModernBERT-FT (canonical)	0.899	36.5
+ F1 Symbolic Aug.	**0.902**	**98.2**
+ all-four Symbolic Aug.	0.893	97.8
Reference		
UGV	0.537	48.1
DeepSeek V4-Flash	0.908	83.4

Table 2: Headline results, 5-fold pooled. The F1-probe column is canonical-equivalence accuracy (%); UGV and a closed-frontier LLM are shown for reference.

Augmentation	F1	F2	F3	F4
none (canonical)	36.5	100	96.7	68.0
+ F1	**98.9**	100	96.7	96.0
+ F2	47.0	100	96.7	28.0
+ F3	33.1	100	96.7	32.0
+ F4	49.7	100	96.7	**96.0**
+ all four	98.9	100	100	96.0

Table 3: Per-family probe accuracy (%), single fold. Rows are training configurations, columns the four probes; training with a family raises its own probe (diagonal, bold).

synonyms (100%), unit-name paraphrase (96.7%), scientific notation (68%)—it is robust or nearly so. On canonical-equivalence rewrites it scores only 36.5%: shown a gold CORRECT summary that states a quantity in a physically equivalent but textually different form (95 °C as 368.15 K), the encoder calls it an error nearly two-thirds of the time. This is the structure hidden in the 0.855 CORRECT score: canonical-equivalent items are a minority of CORRECT—large enough to depress its F_1, too rare for the encoder to learn in-distribution. The failure is not noise; it concentrates on the one transformation that requires canonicalizing a numerical surface rather than matching lexical or syntactic patterns.

5.3 Symbolic Augmentation closes the gap

Training with the F1 (canonical-equivalence) family raises F1-probe accuracy from 36.5% to 98.2% (5-fold pooled; Table 2)—a 62-point gain that closes essentially all of the blind spot—while macro-F_1 rises slightly, 0.899 → 0.902. The robustness gain comes free, not bought with in-distribution accuracy. As Table 3 shows, the F4 (scientific-notation) family similarly lifts its own probe from 68% to 96%, while F2 and F3 yield no gain applied alone—their probes already start

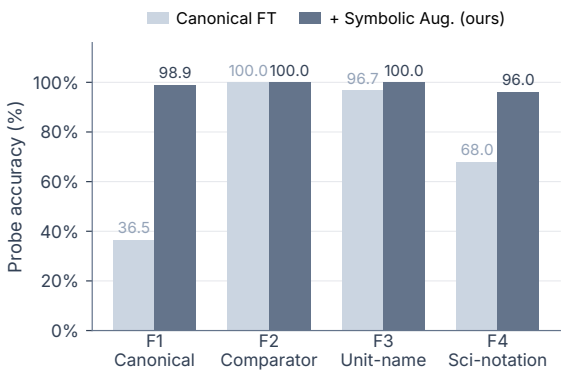

Figure 2: Per-family probe accuracy before and after Symbolic Augmentation. The canonical encoder collapses to 36.5% on canonical-equivalence rewrites (F1); augmentation closes this gap while leaving the other probes near-saturated.

near saturation—so we use them only within the combined set. Figure 2 contrasts the canonical and augmented encoders across all four probes; training with all four families at once raises every probe to at least 96% while holding macro-F_1 at 0.893, statistically indistinguishable from baseline. We therefore carry two variants forward: F1-augmentation as the strongest single intervention, all-four as the broadest. We also observe cross-family transfer—F1 alone lifts the F4 probe from 68% to 96%, so exposure to one numerical-surface rewrite generalizes to another. The blind spot is not unique to small encoders: DeepSeek V4-Flash, a closed-frontier model, scores 83.4% on the same probe—far above the canonical encoder but short of robust. F1-augmented fine-tuning thus *matches* V4-Flash on overall macro-F_1 (0.902 vs. 0.908) and *exceeds* it on canonical-equivalence robustness by nearly 15 points, while running locally at no per-query cost.

Mechanism. Figure 3 traces the F1 probe at the prediction level. All 181 probe variants are gold CORRECT, yet the canonical encoder labels only 66 of them CORRECT. The 115 misclassifications are not spread evenly: 99 of them—86%—land in SCALE ERROR. The encoder, shown a canonical-equivalent rewrite such as 95 °C → 368.15 K, most often reads it as a magnitude that has been altered. A further 15 land in UNIT ERROR; almost none reach RELATION ERROR, confirming that the failure is a *numerical-surface* confusion rather than a relational one. Symbolic Augmentation repairs 114 of the 115 (99.1%): the rewritten CORRECT examples teach the encoder to read the altered surface as

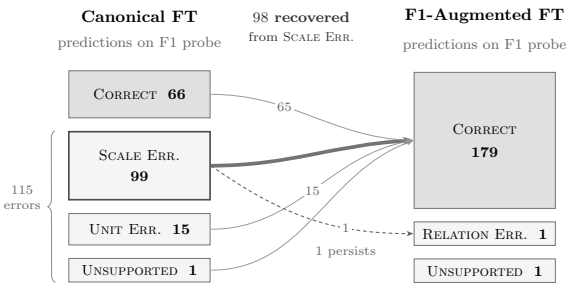

Figure 3: Prediction flow on the F1 probe, all 181 variants gold CORRECT. The canonical encoder (left) sends 99 of its 115 errors to SCALE ERROR; Symbolic Augmentation (right) recovers 114 of the 115.

Integration strategy	**macro-F_1**	Δ
Canonical fine-tuning	0.899	—
Inference-time integration		
Logit ensemble	0.895	-0.004
Agreement filter	0.888	-0.011
Feature-level integration		
UGV output as features	0.879	-0.020
Silver-label training with UGV		
1200 silver items	0.622	-0.277
3298 silver items	0.518	-0.381
Training-data integration (ours)		
Symbolic Augmentation	**0.902**	$\mathbf{+0.003}$

Table 4: Five integration strategies, grouped by where the verifier enters the pipeline, against canonical fine-tuning. Only training-data integration (Symbolic Augmentation) improves macro-F_1; see Appendix H.

a restatement. A single error persists—a milligram-to-gram rewrite the augmented model still scores RELATION ERROR—and one previously correct item is newly misread, for a net movement of $+113$ items into CORRECT.

Out-of-distribution transfer. To test whether the effect is specific to our benchmark, we evaluate on SciFact-Open (Wright et al., 2022), an external scientific claim-verification benchmark, cast as binary verification. Symbolic Augmentation raises binary macro-F_1 from 0.791 to 0.828 (+0.037) without any SciFact-Open training data, extending a monotone ladder that runs from trivial baselines through rule-based systems and UGV to the fine-tuned encoders. The improvement confirms that augmentation instills a transferable capability—canonicalizing numerical surfaces—rather than a benchmark-specific pattern. The full ladder and per-class results are in Appendix I.

5.4 Where symbolic information helps

Symbolic Augmentation integrates the verifier at the training-data layer. Four other integration points are equally plausible a priori, yet Table 4 shows none works. At inference time, a logit ensemble with UGV (-0.004) is within noise, and an agreement filter—take the joint prediction when UGV and the encoder agree, else the encoder—reduces mathematically to the encoder alone. At the feature level, concatenating the verifier's output as a 12-dimensional auxiliary vector—a one-hot UGV prediction plus continuous module-trace signals—costs 0.020 macro-F_1: the encoder already extracts what the features encode. Using the verifier as a silver-label teacher is actively harmful and worsens with scale (0.622 at 1200 items, 0.518 at 3298), because the verifier's labels are too noisy—it agrees with the gold label on only 53.1% of items—and inject error faster than volume compensates. Only training-time augmentation both improves macro-F_1 and delivers the 62-point probe gain, because it is the one integration point that inherits the verifier's deterministic label guarantee without inheriting its accuracy ceiling. A full per-method analysis is in Appendix H.

6 Conclusion

We framed the verification of numbers and units in LLM-generated scientific text as *typed* classification and released a five-class taxonomy and a 1500-item benchmark to study it. A fine-tuned ModernBERT encoder reaches $macro - F_1 = 0.899$ on this benchmark, far beyond any off-the-shelf neural fact-checker, yet four probes expose a sharp blind spot: it scores only 36.5% on canonical-equivalent rewrites of physically equivalent quantities. Our main contribution, **Symbolic Augmentation**, closes this gap by transferring the modules of a symbolic verifier into the encoder as training-time data augmentation, raising canonical-equivalence robustness to 98.2% while slightly improving in-distribution accuracy, matching a closed-frontier LLM at no inference cost, and transferring to an external benchmark. Of five ways to combine a symbolic verifier with a learned model, only training-time augmentation delivers a transferable benefit. We hope the benchmark and framework support further work on bringing symbolic guarantees into learned verifiers.

Limitations

Three scope conditions bound our results. First, the benchmark is built from English-language PMC and arXiv sources; the taxonomy is defined over language-independent symbolic properties and is deliberately dataset-agnostic, and we verify cross-dataset transfer on SciFact-Open (§5.3), but other languages remain future work. Second, Symbolic Augmentation requires a symbolic verifier whose modules implement deterministic, label-preserving rewrites; the framework is general with respect to such verifiers, though constructing them for new domains needs additional engineering. Third, gold labels are LLM-produced; we mitigate the self-labeling concern with a disjoint probe generator and a 50-item human spot-check that agrees with the gold labels on 94% of items.

References

Rami Aly and Andreas Vlachos. 2024. Tabver: Tabular fact verification with natural logic. *Transactions of the Association for Computational Linguistics*, 12:1648–1671.

Abhijit Anand, Avishek Anand, Vinay Setty, and 1 others. 2024. Quantemp: A real-world open-domain benchmark for fact-checking numerical claims. *arXiv preprint arXiv:2403.17169*.

Anirban Saha Anik, Md Fahimul Kabir Chowdhury, Andrew Wyckoff, and Sagnik Ray Choudhury. 2025. Claimiq at checkthat! 2025: comparing prompted and fine-tuned language models for verifying numerical claims. *arXiv preprint arXiv:2509.11492*.

Akari Asai, Jacqueline He, Rulin Shao, Weijia Shi, Amanpreet Singh, Joseph Chee Chang, Kyle Lo, Luca Soldaini, Sergey Feldman, Mike D'arcy, and 1 others. 2024. Openscholar: Synthesizing scientific literature with retrieval-augmented lms. *arXiv preprint arXiv:2411.14199*.

Yejin Bang, Ziwei Ji, Alan Schelten, Anthony Hartshorn, Tara Fowler, Cheng Zhang, Nicola Cancedda, and Pascale Fung. 2025. Hallulens: Llm hallucination benchmark. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 24128–24156.

Yonatan Belinkov. 2022. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219.

Bespoke Labs. 2024. Bespoke-MiniCheck-7B. https://huggingface.co/bespokelabs/Bespoke-MiniCheck-7B.

Daniil A Boiko, Robert MacKnight, Ben Kline, and Gabe Gomes. 2023. Autonomous chemical research with large language models. *Nature*, 624(7992):570–578.

Jiaao Chen, Derek Tam, Colin Raffel, Mohit Bansal, and Diyi Yang. 2023. An empirical survey of data augmentation for limited data learning in nlp. *Transactions of the Association for Computational Linguistics*, 11:191–211.

I Chern, Steffi Chern, Shiqi Chen, Weizhe Yuan, Kehua Feng, Chunting Zhou, Junxian He, Graham Neubig, Pengfei Liu, and 1 others. 2023. Factool: Factuality detection in generative ai–a tool augmented framework for multi-task and multi-domain scenarios. *arXiv preprint arXiv:2307.13528*.

Primakov Chungkham, Venktesh Viswanathan, Vinay Setty, and Avishek Anand. 2025. Think right, not more: Test-time scaling for numerical claim verification. In *Empirical Methods in Natural Language Processing (EMNLP) 2025*, pages 24345–24363. Association for Computational Linguistics.

Zahra Delbari and Mohammad Taher Pilehvar. 2025. Beyond accuracy: Revisiting out-of-distribution generalization in nli models. In *Proceedings of the 29th Conference on Computational Natural Language Learning*, pages 557–570.

Ruixiang Feng, Zhenwei An, Yuntao Wen, Ran Le, Yiming Jia, Chen Yang, Zongchao Chen, Lisi Chen, Shen Gao, Shuo Shang, and 1 others. 2025. Cosineverifier: Tool-augmented answer verification for computation-oriented scientific questions. *arXiv preprint arXiv:2512.01224*.

John M Ford. 2004. Content analysis: An introduction to its methodology. *Personnel psychology*, 57(4):1110.

Hernán E Grecco and 1 others. 2024. Pint: A Python units library. https://github.com/hgrecco/pint.

Maximilian Heil and Aleksandar Pramov. 2025. Ds@gt at checkthat! 2025: evaluating context and tokenization strategies for numerical fact verification. *arXiv preprint arXiv:2507.06195*.

Yuelyu Ji, Min Gu Kwak, Hang Zhang, Xizhi Wu, Chenyu Li, and Yanshan Wang. 2026. Medragchecker: Claim-level verification for biomedical retrieval-augmented generation. *arXiv preprint arXiv:2601.06519*.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM computing surveys*, 55(12):1–38.

Deren Lei, Yaxi Li, Siyao Li, Mengya Hu, Rui Xu, Ken Archer, Mingyu Wang, Emily Ching, and Alex Deng.

2025. Factcg: Enhancing fact checkers with graph-based multi-hop data. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5002–5020.

Weixin Liang, Zachary Izzo, Yaohui Zhang, Haley Lepp, Hancheng Cao, Xuandong Zhao, Lingjiao Chen, Haotian Ye, Sheng Liu, Zhi Huang, and 1 others. 2024. Monitoring ai-modified content at scale: A case study on the impact of chatgpt on ai conference peer reviews. *arXiv preprint arXiv:2403.07183*.

Andres M. Bran, Sam Cox, Oliver Schilter, Carlo Baldassari, Andrew D White, and Philippe Schwaller. 2024. Augmenting large language models with chemistry tools. *Nature machine intelligence*, 6(5):525–535.

Potsawee Manakul, Adian Liusie, and Mark Gales. 2023. 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.

Maggie Mi, Aline Villavicencio, and Nafise Sadat Moosavi. 2025. From input perception to predictive insight: Modeling model blind spots before they become errors. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 34316–34329.

Swaroop Mishra, Arindam Mitra, Neeraj Varshney, Bhavdeep Sachdeva, Peter Clark, Chitta Baral, and Ashwin Kalyan. 2022. Numglue: A suite of fundamental yet challenging mathematical reasoning tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3505–3523.

Vaibhav Mishra, Somaditya Singh, Dhruv Ahlawat, Mohd Zaki, Vaibhav Bihani, Hargun Singh Grover, Biswajit Mishra, Santiago Miret, NM Krishnan, and 1 others. 2024. Foundational large language models for materials research. *arXiv preprint arXiv:2412.09560*.

Inkit Padhi, Manish Nagireddy, Giandomenico Cornacchia, Subhajit Chaudhury, Tejaswini Pedapati, Pierre Dognin, Keerthiram Murugesan, Erik Miehling, Martín Santillán Cooper, Kieran Fraser, and 1 others. 2024. Granite guardian. *arXiv preprint arXiv:2412.07724*.

Rishanth Rajendhran, Amir Zadeh, Matthew Sarte, Chuan Li, and Mohit Iyyer. 2025. Verifastscore: Speeding up long-form factuality evaluation. *arXiv preprint arXiv:2505.16973*.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of nlp models with checklist. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 4902–4912.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in bertology: What we know about how bert works. *Transactions of the association for computational linguistics*, 8:842–866.

Wooseok Seo, Seungju Han, Jaehun Jung, Benjamin Newman, Seungwon Lim, Seungbeen Lee, Ximing Lu, Yejin Choi, and Youngjae Yu. 2025. Verifying the verifiers: Unveiling pitfalls and potentials in fact verifiers. *arXiv preprint arXiv:2506.13342*.

Damien Sileo. 2024. tasksource: A large collection of NLP tasks with a structured dataset preprocessing framework. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 15655–15684, Torino, Italia. ELRA and ICCL.

Amanpreet Singh, Joseph Chee Chang, Dany Haddad, Aakanksha Naik, Jena D Hwang, Rodney Kinney, Daniel S Weld, Doug Downey, and Sergey Feldman. 2025. Ai2 scholar qa: Organized literature synthesis with attribution. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 513–523.

Hwanjun Song, Hang Su, Igor Shalyminov, Jason Cai, and Saab Mansour. 2024. Finesure: Fine-grained summarization evaluation using llms. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 906–922.

Liyan Tang, Philippe Laban, and Greg Durrett. 2024. Minicheck: Efficient fact-checking of llms on grounding documents. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8818–8847.

Avijit Thawani, Jay Pujara, Filip Ilievski, and Pedro Szekely. 2021. Representing numbers in nlp: a survey and a vision. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–656.

The quantulum3 contributors. 2024. quantulum3: A Python library for quantity extraction from unstructured text. https://github.com/nielstron/quantulum3.

Bhanu Prakash Vangala, Sajid Mahmud, Pawan Neupane, Joel Selvaraj, and Jianlin Cheng. 2025. Hallumat: Detecting hallucinations in llm-generated materials science content through multi-stage verification. *arXiv preprint arXiv:2512.22396*.

Raúl Vázquez, Timothee Mickus, Elaine Zosa, Teemu Vahtola, Jörg Tiedemann, Aman Sinha, Vincent Segonne, Fernando Sánchez-Vega, Alessandro Raganato, Jindřich Libovickỳ, and 1 others. 2025. Semeval-2025 task 3: Mu-shroom, the multilingual shared-task on hallucinations and related observable overgeneration mistakes. In *Proceedings of the*

19th International Workshop on Semantic Evaluation (SemEval-2025), pages 2472–2497.

V Venktesh, Deepali Prabhu, and Avishek Anand. 2025a. A benchmark for open-domain numerical fact-checking enhanced by claim decomposition. *arXiv preprint arXiv:2510.22055*.

V Venktesh, V Setty, A Anand, M Hasanain, B Bendou, H Bouamor, F Alam, G Iturra-Bocaz, and P Galuščáková. 2025b. Overview of the clef-2025 checkthat! lab task 3 on fact-checking numerical claims. *Working Notes of CLEF*.

David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020. Fact or fiction: Verifying scientific claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7534–7550.

David Wadden and Kyle Lo. 2021. Overview and insights from the sciver shared task on scientific claim verification. In *Proceedings of the Second Workshop on Scholarly Document Processing*, pages 124–129.

David Wan, Koustuv Sinha, Srini Iyer, Asli Celikyilmaz, Mohit Bansal, and Ramakanth Pasunuru. 2024. Acueval: Fine-grained hallucination evaluation and correction for abstractive summarization. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 10036–10056.

Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, and 1 others. 2025. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2526–2547.

Chenxi Whitehouse, Monojit Choudhury, and Alham Fikri Aji. 2023. Llm-powered data augmentation for enhanced cross-lingual performance. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pages 671–686.

Dustin Wright, David Wadden, Kyle Lo, Bailey Kuehl, Arman Cohan, Isabelle Augenstein, and Lucy Lu Wang. 2022. Generating scientific claims for zero-shot scientific fact checking. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2448–2460.

Luhao Zhang, Xinyu Zhang, Linmei Hu, Dandan Song, and Liqiang Nie. 2025. Dually self-improved counterfactual data augmentation using large language model. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5216–5227.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, and 1 others. 2023. Siren's song in the ai ocean: A survey on hallucination in large language models. arxiv 2023. *arXiv preprint arXiv:2309.01219*.

Yingjie Zhu, Jiasheng Si, Yibo Zhao, Haiyang Zhu, Deyu Zhou, and Yulan He. 2023. Explain, edit, generate: rationale-sensitive counterfactual data augmentation for multi-hop fact verification. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13377–13392.

A Task Details

Priority resolution. A single rewritten summary can exhibit more than one error type at once—for instance, it may both mis-state a magnitude and add an unsupported superlative. In such cases the gold label is the single highest-priority error present, resolved by the strict ordering UNSUPPORTED $\succ$ UNIT ERROR $\succ$ SCALE ERROR $\succ$ RELATION ERROR $\succ$ CORRECT. The ordering reflects two principles. First, UNSUPPORTED ranks above every typed numeric error because it marks an *epistemic* shift independent of the numeric content: an unsupported causal or superlative claim misleads the reader whether or not its quantities are correct. Second, among the three typed numeric errors the ordering follows *axis precedence*: a dimensional mismatch (UNIT ERROR) is more fundamental than a magnitude mismatch (SCALE ERROR), which is in turn more fundamental than a relational mismatch (RELATION ERROR). The ordering is empirically active on 116 of the 1500 benchmark items (7.7%): in each, the rewriter was prompted for a typed numeric error but additionally introduced a superlative, causal, or significance phrase, and the annotators relabeled the item UNSUPPORTED.

Repair. Beyond assigning a label, a verifier may optionally emit a *templated repair* r: a minimal edit that, applied to the summary, would render it CORRECT with respect to the evidence. The repair template depends on the error type:

- For UNIT ERROR and SCALE ERROR, the repair substitutes the evidence's canonical quantity for the summary's incorrect one.
- For RELATION ERROR, the repair restores the evidence's original comparator direction or entity–value binding.
- For UNSUPPORTED, the repair deletes the unsupported span.

Repair is a secondary capability, not a headline metric: the systems we focus on—the fine-tuned encoder and its augmented variants—are classifiers and emit no repair. UGV produces a repair as a byproduct of its symbolic trace, since each firing stage already localizes the offending span; we report UGV's repair accuracy only for completeness.

B The UNITGRAPH VERIFIER Pipeline

UNITGRAPH VERIFIER (UGV) is a deterministic, four-stage verification pipeline. Algorithm 1 states it in full; this section describes each stage and the symbolic checks it performs.

Stage 1: extraction and normalization. The pipeline first extracts every quantity mention from both the evidence and the summary using quantulum, which returns, for each mention, a tuple of numeric value, unit, and character span. A hand-curated normalizer then canonicalizes these tuples: it maps unit aliases to a single surface form (for example `M` to `mol/L` and `wt%` to percent), expands scientific notation to a plain decimal, and resolves context-implicit units. Normalization ensures that two physically equivalent quantities written differently are compared on equal footing in the later stages.

Stage 2: dimensional and magnitude checks. After normalization, each summary quantity q_s is aligned with its corresponding evidence quantity q_e by matching entity head and modified property. For each aligned pair, UGV applies two checks. The *dimensional check* compares physical dimensions: if

$$\dim(q_s) \neq \dim(q_e),$$

the pair is a dimensional mismatch and the pipeline returns UNIT ERROR. Otherwise the *magnitude check* compares canonicalized values on a logarithmic scale: if

$$\left|\log_{10}(q_s/q_e)\right| \geq 1,$$

the two values differ by at least one order of magnitude and the pipeline returns SCALE ERROR. The magnitude check is suppressed when the two surface forms canonicalize to the same physical quantity within 1%, which prevents a canonical-equivalent restatement from being misreported as a magnitude error.

Stage 3: relational checks. If no quantity pair triggers a dimensional or magnitude error, UGV checks relational structure. The COMPARATORFLIP test detects a reversed comparison: it tags comparator words in both sentences with a polarity drawn from curated clusters (the "high" cluster contains *higher*, *elevated*, *greater*, and so on; the "low" cluster its antonyms) and fires when matched comparators carry opposite polarity. The ENTITYSWAP test detects a quantity bound to the wrong entity.

Algorithm 1 UGV verification pipeline

Require: evidence e, summary s
Ensure: label $y \in \{$CORRECT, UNIT ERROR, SCALE ERROR, RELATION ERROR, UNSUPPORTED $\}$

```
T ← TRIGGERS(s) \ TRIGGERS(e)
if T ≠ ∅ then return UNSUPPORTED                                  ▷ Stage 4
end if
Q_e, Q_s ← EXTRACT(e), EXTRACT(s)
Q_e, Q_s ← NORMALIZE(Q_e), NORMALIZE(Q_s)
for (q_e, q_s) in ALIGN(Q_e, Q_s) do
    if dim(q_s) ≠ dim(q_e) then return UNIT ERROR                 ▷ Stage 2
    end if
    if |log_10(q_s/q_e)| ≥ 1 then return SCALE ERROR              ▷ Stage 2
    end if
end for
if COMPARATORFLIP(e, s) or ENTITYSWAP(e, s) then return RELATION ERROR   ▷ Stage 3
end if
return CORRECT
```

If either test fires, the pipeline returns RELATION ERROR.

Stage 4: unsupported-claim triggers. Although highest in priority and therefore executed first, this stage checks whether the summary introduces a claim absent from the evidence. UGV maintains an inventory of trigger phrases in three groups:

Superlative highest, lowest, largest, smallest, best, worst, unprecedented, record.

Causal causes, leads to, results in, due to, responsible for, drives, dominant factor.

Significance statistically significant, significant improvement, significantly, $p < 0.05$, demonstrates.

The pipeline computes the set of trigger phrases occurring in the summary but not in the evidence; if this set is non-empty, it returns UNSUPPORTED. Matching is case-insensitive and operates over lemmatized tokens, so that inflectional variants (*cause*, *causes*, *caused*) are treated alike. The five significance-group phrases are reused, as the only unsupported triggers, by the unit-substring and quantulum baselines described in Appendix E.

C Probe Construction

Each augmentation family F_i is defined by a set of deterministic, label-preserving rewrite rules over quantity surfaces. The per-family *probe* is built directly from these rules, so that a probe and its corresponding training augmentation exercise exactly the same transformation—they call the same rewrite code.

Probe construction proceeds as follows. We take the 333 test-set items whose gold label is CORRECT. For each item, we apply the family's rewrite rules to the summary, producing one or more *variants* in which a quantity has been re-expressed in a physically equivalent but textually different form. Because every rewrite rule is label-preserving by construction, each variant still has gold label CORRECT. The probe for family F_i is the resulting set of variants V_i, and the model is scored on whether it continues to assign them CORRECT.

The four probes contain

$$|V_1| = 181, \quad |V_2| = 22, \quad |V_3| = 30, \quad |V_4| = 25$$

variants respectively. The counts differ because a family can only rewrite an item that contains an applicable quantity: F4 (scientific-notation rewriting), for example, applies only to items whose summary contains a number that can be re-expressed in scientific notation, and comparatively few CORRECT items qualify.

D Dataset Construction

This section details the four-step construction pipeline: source-sentence sampling, LLM rewriting, two-annotator labeling with adjudication, and a human spot-check. Figure 4 summarizes the labeling flow.

Source sampling. Evidence sentences are drawn from three scientific domains, each contributing roughly one third of the benchmark:

- **PMC biomedical** (484 sentences). We retrieve approximately 120 open-access articles through the NCBI E-utilities API, using a query targeting concentration and pharmacokinetic terminology (for example `mg/mL`, `µM`, `IC50`, *pharmacokinetic*), then extract sentences at the sentence level.

- **PMC materials science** (507 sentences). A parallel retrieval of roughly 120 open-access articles uses a query targeting battery and photovoltaic terminology (for example *perovskite*, *cathode*, `mAh/g`, *photocurrent*).

- **arXiv computer science** (509 sentences). Sentences are retrieved through the Semantic Scholar and arXiv APIs with queries targeting evaluation and benchmarking terminology.

A candidate sentence is retained only if quantulum can extract at least one quantity from it, ensuring every evidence sentence has quantitative content. Evidence sentences are further restricted to at most three quantity tuples, which keeps quantity alignment unambiguous in later processing. The 1500 benchmark items derive from 741 unique evidence sentences, so each evidence yields roughly two rewritten summaries on average.

Rewriting. Each evidence sentence is rewritten into candidate summaries by claude-haiku-4-5 (maximum 500 output tokens). The system prompt supplies the complete five-class definitions together with per-class strategy guidelines describing how to construct each error type; the user prompt supplies one evidence sentence and one target label, and requests a single fluent summary that, compared against the evidence, exhibits the targeted class. The model returns the summary wrapped in explicit delimiters. We rewrite each evidence against multiple target labels, cycling through classes until each class has accumulated 300 candidates, for 1500 candidates in total.

Annotation and adjudication. Every candidate summary is then independently labeled by two annotators drawn from *different model families*, which is central to avoiding a self-labeling artifact:

- **Annotator A**: claude-haiku-4-5 (Anthropic).

- **Annotator B**: gpt-5.4-mini (OpenAI).

- **Adjudicator**: claude-opus-4-7 (Anthropic), invoked only on items where A and B disagree.

Each annotator receives the evidence, the candidate summary, and the five-class definitions, and returns a single label. The two annotators agree on 1360 of 1500 items; on the remaining 140 (9.3%) the adjudicator assigns the final label. Inter-annotator agreement is measured by Krippendorff's α computed over annotators A and B only:

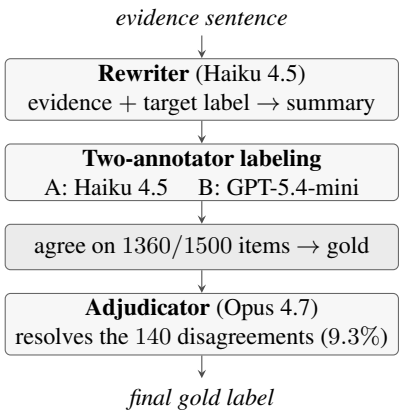

Figure 4: The annotation pipeline. Each rewritten summary is labeled independently by two annotators from different model families; the 9.3% of items on which they disagree are resolved by an adjudicator. Krippendorff's α is computed over the two annotators only.

$$\alpha = 1 - \frac{D_o}{D_e} = 0.882,$$

where D_o and D_e are the observed and expected disagreement. The adjudicator does not enter this computation. Because annotator B belongs to a different model family than the rewriter (claude-haiku-4-5), this α does not reflect a model scoring its own output, and the value comfortably exceeds the conventional 0.80 threshold for reliable annotation.

Gold label distribution. The rewriter targets a balanced 300 items per class, but adjudication shifts the final distribution, because the priority rule of Appendix A relabels some items. The gold distribution is:

Class	Gold count
CORRECT	333
UNIT ERROR	236
SCALE ERROR	240
RELATION ERROR	290
UNSUPPORTED	401
Total	**1500**

The final distribution departs from the balanced 300-per-class target for two reasons: the priority rule of Appendix A relabels 116 items to UNSUPPORTED, and annotators sometimes judged a rewrite to realize a different class than the one targeted. Both effects move mass toward UNSUPPORTED and CORRECT and away from UNIT ERROR and SCALE ERROR. The three source domains remain near-balanced, contributing 484, 507, and 509 items.

Human spot-check. To validate the LLM gold labels against a human standard, one of the authors—blind to all model predictions and to the paper's analysis—independently labeled a random 50-item sample. The human label matches the adjudicated gold label on 47 of 50 items (94%). All three disagreements are SCALE ERROR-versus-RELATION ERROR boundary cases involving sub-ten-fold numerical differences, a region in which the LLM annotators also reported low confidence; they are genuine borderline cases rather than annotation errors.

E Baseline Configurations

We evaluate five groups of baseline systems. This section specifies each one: its checkpoint or definition, its native output, and—for systems whose native output is not already in our label space—how that output is routed to a five-class label.

Trivial baselines. These establish the floor of the metric:

predict-CORRECT returns CORRECT for every item, unconditionally. It measures the score obtainable with no modeling at all, and coincides numerically with a majority-class predictor, since CORRECT is the majority class of the development split.

string-overlap returns CORRECT if the evidence and summary are identical after whitespace trimming, and UNSUPPORTED otherwise. It uses exact matching, with no similarity threshold.

unit-substring extracts each value-and-unit token from the summary with a regular expression. It first checks for the five significance-group trigger phrases (Appendix B) and returns UNSUPPORTED if any is present; otherwise it returns UNIT ERROR if any extracted token is absent from the evidence, and CORRECT if all tokens are present.

Rule-based baseline.

quantulum + Pint extracts quantity tuples from both sentences with quantulum, position-aligns them, and applies the Stage 2 checks of Appendix B: a dimensional mismatch yields UNIT ERROR and an order-of-magnitude difference yields SCALE ERROR. It performs no comparator or entity-binding analysis, which is the capability UGV adds in Stage 3.

Our symbolic verifier.

UGV is the full four-stage pipeline of Appendix B. It uses neither task supervision nor a language model at inference time.

Off-the-shelf neural fact-checkers. Each of the following emits a binary or ternary entailment decision natively. We list each system with its checkpoint and native output:

MiniCheck (lytang/MiniCheck-Flan-T5-Large): a Flan-T5 model emitting a "Yes"/"No" supported decision.

FactCG (yaxili96/FactCG-DeBERTa-v3-Large): a DeBERTa-v3 classifier whose positive label denotes "supported."

ModernBERT-NLI, zero-shot (tasksource/ModernBERT-large-nli): a three-way NLI model; we read the "entailment" class as supported.

Bespoke-MiniCheck-7B (bespokelabs/Bespoke-MiniCheck-7B): a 7B instruction-tuned model emitting a "Yes"/"No" decision.

Granite Guardian (ibm-granite/granite-guardian-3.3-8b): an 8B groundedness judge emitting a "Yes"/"No" decision via its chat template.

VeriFastScore (lytang/VeriFastScore-8B): an 8B high-throughput verifier that emits a binary supported decision.

Label routing. Because these systems emit only a support decision, we route their output into the five-class space with a fixed rule. A "supported" or "entailment" output maps directly to CORRECT. For any non-supported output, we apply the following cascade:

1. if the summary carries an unsupported trigger phrase absent from the evidence, return UNSUPPORTED;
2. else if a comparator flip is detected between summary and evidence, return RELATION ERROR;
3. else return UNIT ERROR as the default.

This cascade has a structural consequence: it can never output SCALE ERROR. This is why every off-the-shelf neural baseline scores exactly 0.000 on SCALE ERROR in Table 1—a limitation of the binary-entailment paradigm, not a tuning artifact.

Closed-frontier LLMs. We evaluate three closed-weight models—Claude Opus 4.7, GPT-5.5, and DeepSeek V4-Flash—accessed through their respective APIs. All three are run zero-shot with chain-of-thought prompting and greedy decoding (temperature 0). The system prompt contains the full five-class definitions and four in-prompt example triples, each a (evidence, summary, label) example for a distinct class; the user prompt supplies the evidence and summary and requests a label. Output token budgets are 512 for Claude and GPT and 1024 for DeepSeek V4-Flash, which requires extra space for reasoning tokens. All three are evaluated on the full 1500-item benchmark.

F Metrics

This section gives the precise definition of every metric used in the paper.

Macro-F_1. For each class c, let TP_c, FP_c, and FN_c denote true positives, false positives, and false negatives. The per-class precision and recall are

$$P_c = \frac{\mathrm{TP}_c}{\mathrm{TP}_c + \mathrm{FP}_c}, \qquad R_c = \frac{\mathrm{TP}_c}{\mathrm{TP}_c + \mathrm{FN}_c},$$

and the per-class F_1 is their harmonic mean,

$$F_{1,c} = \frac{2\, P_c\, R_c}{P_c + R_c}.$$

The primary metric, macro-F_1, is the unweighted mean over the five classes,

$$F_1^{\mathrm{macro}} = \frac{1}{5} \sum_{c \in \mathcal{C}} F_{1,c},$$

where $\mathcal{C}$ ranges over the five classes CORRECT, UNIT ERROR, SCALE ERROR, RELATION ERROR, and UNSUPPORTED. The unweighted mean gives the rare and common classes equal weight, so a system cannot score well merely by serving the majority class.

Probe accuracy. For augmentation family F_i with probe variant set V_i (Appendix C), every variant $v \in V_i$ has gold label CORRECT. Probe accuracy is the fraction the model $\hat{y}$ still classifies CORRECT,

$$\mathrm{acc}_{F_i} = \frac{1}{|V_i|} \sum_{v \in V_i} \mathbb{1}[\, \hat{y}(v) = \textsc{Correct}\,],$$

where $\mathbb{1}[\cdot]$ is the indicator function. Since every probe variant is a label-preserving rewrite of a genuinely CORRECT item, probe accuracy is exactly the model's recall on that family's surface transformation.

Out-of-distribution metric. SciFact-Open provides binary SUPPORT/CONTRADICT labels. To apply our five-class models to it, we fold each prediction y with the mapping

$$\phi(y) = \begin{cases} \text{SUPPORT} & \text{if } y = \textsc{Correct}, \\ \text{CONTRADICT} & \text{otherwise}, \end{cases}$$

that is, the four error classes all map to CONTRADICT. We then report binary macro-F_1, the unweighted mean of the SUPPORT and CONTRADICT F_1 scores. The out-of-distribution evaluation is detailed in Appendix I.

G Training Details

Model and optimization. The fine-tuned model uses tasksource/ModernBERT-large-nli (400M parameters) as its backbone. The pretrained three-class NLI head is discarded and replaced with a randomly initialized five-class linear classifier; the size mismatch between the old and new heads is handled by allowing mismatched checkpoint sizes at load time. The full optimization configuration is:

- **Optimizer**: AdamW, learning rate 2×10^{-5}, weight decay 0.01.
- **Schedule**: linear warmup over the first 10% of steps, followed by linear decay.
- **Batch size**: 8, limited by GPU memory at sequence length 512.
- **Epochs**: 3, with evaluation at the end of training and no early stopping.
- **Sequence length**: 512 tokens, with dynamic padding within each batch.
- **Loss**: cross-entropy.
- **Gradient clipping**: maximum norm 1.0.
- **Precision**: float32, with no mixed precision.

Hardware. Each fold is trained on a single NVIDIA RTX 3090 GPU (24GB). A single fold completes in two to three minutes, and the full five-fold run completes in twelve to fifteen minutes of wall-clock time.

Cross-validation. We use stratified five-fold cross-validation, stratified by gold label, with 1200 training and 300 test items per fold. We report two evaluation modes. The *5-fold pooled* mode—used for all headline numbers—pools the out-of-fold predictions across all five folds and computes

macro-F_1 on the full 1500 items. The *single-fold* mode uses one fixed 80/20 stratified split. Pooled numbers are more stable, since per-fold single-split numbers fluctuate with the particular split.

Augmentation volume. Symbolic Augmentation is applied only to the gold CORRECT items of each training fold, so the label distribution of the augmented training set matches the original. F1 augmentation adds approximately 155 canonical-equivalent items per fold (its rewrite rules applied to the fold's CORRECT items, capped at three new items per source item), giving a combined training set of roughly 1350 items. All-four augmentation adds approximately 206 items per fold (capped at four per source item across the four families), giving roughly 1406 items.

H Negative-Result Setups

This section specifies the configuration of each integration point compared in Table 4.

Auxiliary symbolic features. This setup concatenates the symbolic verifier's output to the encoder as an auxiliary input. The UGV output is encoded as a 12-dimensional vector: a 5-dimensional one-hot of the UGV predicted label, plus a 7-dimensional continuous trace of intermediate signals—the log-scaled evidence and summary quantity counts, a dimensional-mismatch flag, a comparator-flip flag, a superlative-trigger flag, a scale-gap log-ratio clamped to $[-3, 3]$, and an entity-swap flag. Architecturally, the ModernBERT encoder output is mean-pooled over non-padding tokens, concatenated with a 64-dimensional learned projection of the auxiliary vector, and passed through a two-layer classification head. Training otherwise matches the canonical fine-tuning configuration of Appendix G. To isolate the contribution of the auxiliary features, we train two variants that are identical except that one receives the true feature vector and the other receives a zeroed vector.

Silver-label teachers. This setup uses UGV as a teacher that labels training data, which is then used to train the encoder. We evaluate it at two scales:

- $n = 1200$: the gold labels of the canonical training split are replaced with UGV's predicted labels. UGV agrees with the gold label on 53.1% of these items, so the silver set is substantially noisier than gold.
- $n = 3298$: in addition, five new perturbations per evidence are generated by gpt-5.4-mini with the same five-class taxonomy prompt as the main rewriter, and each is labeled by UGV. After deduplication against existing test summaries, 3298 silver items remain.

In both cases the silver set replaces the gold training set entirely; the encoder is otherwise trained as in Appendix G.

Inference-time logit ensemble. This setup keeps the canonical encoder unchanged and combines its output with UGV's at inference time. Since UGV is rule-based and emits a hard label rather than a distribution, its prediction is represented as a one-hot vector. The ensemble distribution is the convex combination

$$p = (1 - \alpha)\, p_{\mathrm{FT}} + \alpha \cdot \mathrm{onehot}(\hat{y}_{\mathrm{UGV}}),$$

where p_{FT} is the encoder's softmax distribution and $\alpha = 0.25$. The final prediction is $\arg\max_c p_c$.

Agreement filter. This setup uses UGV's prediction only as a gate. For each item, if UGV and the encoder predict the same label, that label is returned; if they disagree, the encoder's prediction is returned. Formally the output is

$$\hat{y} = \begin{cases} \hat{y}_{\mathrm{UGV}} & \text{if } \hat{y}_{\mathrm{UGV}} = \hat{y}_{\mathrm{FT}}, \\ \hat{y}_{\mathrm{FT}} & \text{otherwise.} \end{cases}$$

In the first branch $\hat{y}_{\mathrm{UGV}} = \hat{y}_{\mathrm{FT}}$, so the output equals $\hat{y}_{\mathrm{FT}}$; in the second branch it equals $\hat{y}_{\mathrm{FT}}$ by definition. The filter therefore returns $\hat{y}_{\mathrm{FT}}$ on every item and is mathematically identical to the encoder alone, which is why its macro-F_1 exactly matches the canonical baseline.

I Out-of-Distribution Evaluation Details

Evaluation set. We evaluate out-of-distribution transfer on SciFact-Open (Wright et al., 2022). We load the test split and keep the 206 items carrying a SUPPORT or CONTRADICT label—105 and 101 respectively—discarding items labeled not-enough-information. No further filtering by domain, quantity content, or sentence length is applied. Both fine-tuned models are evaluated zero-shot: they are trained only on the typed-quantity benchmark and never observe SciFact-Open data or its label space, so their five-class predictions are folded to binary using the mapping ϕ of Appendix F.

System	macro-F_1	Acc.
predict-CORRECT	0.338	0.510
string-overlap	0.329	0.490
unit-substring	0.390	0.534
quantulum + Pint	0.427	0.549
UGV	0.455	0.539
ModernBERT-FT (canonical)	0.791	0.796
+ F1 Symbolic Aug.	**0.828**	**0.830**

Table 5: Out-of-distribution results on SciFact-Open ($n = 206$), for the same rule-based and fine-tuned systems as the main experiments.

Model	Class	P	R	F_1
canonical FT	SUPPORT	0.957	0.629	0.759
	CONTRADICT	0.715	0.970	0.824
+ F1 Aug.	SUPPORT	0.938	0.714	0.811
	CONTRADICT	0.762	0.951	0.846

Table 6: Per-class out-of-distribution results on SciFact-Open. Symbolic Augmentation lifts SUPPORT recall by 8.5 points, the main source of the macro-F_1 gain.

Ladder of systems. Table 5 places the two fine-tuned models on a ladder with the same rule-based and symbolic systems used in the main experiments. Performance increases monotonically along the ladder: from the trivial baselines, through the rule-based systems and UGV, to the canonical fine-tuned encoder, and finally to the F1-augmented encoder. Symbolic Augmentation adds $+0.037$ binary macro-F_1 on top of canonical fine-tuning, entirely through zero-shot transfer.

Per-class breakdown. Table 6 breaks the two fine-tuned models down by class. The improvement is concentrated in SUPPORT recall, which rises from 0.629 to 0.714, a gain of 8.5 points. The augmented encoder recognizes more SciFact-Open claims that paraphrase the evidence with canonical-equivalent units as genuinely supporting the evidence, rather than mistaking them for relational errors or unsupported assertions. Concretely, of the 11 items the augmented model newly labels CORRECT—hence SUPPORT—seven had been assigned RELATION ERROR and four UNSUPPORTED by the canonical model. This is the same capability the F1 probe measures in distribution, now visible on an external benchmark.