Proof-Carrying Answers: Machine-Verifiable AI Outputs with Typed Claims and Structured Evidence Chains

Connor J. Frank

Detent.ai

March 2026

Abstract

Large language models produce fluent text but offer no machine-checkable evidence that their outputs are grounded in source material. Confidence scores are not verifiable evidence. We introduce *proof-carrying answers* (PCAs)—structured AI outputs in which every claim is typed, bound to span-level evidence, verified by natural language inference (NLI), and packaged with a digitally signed verification certificate. A PCA bundles sufficient evidence for a downstream consumer to audit each claim's grounding without access to the originating model.

We describe the architecture behind PCAs—claim-first generation with multi-tier NLI verification, XGBoost multi-signal aggregation, and deterministic deflection—and evaluate it on seven attribution and fact-verification benchmarks (with two evaluation protocols for SciFact). Using fine-tuned DeBERTa models [He et al., 2023] (184M–435M parameters), our system achieves 95.3% F1 on SciFact, 94.7% F1 on FEVER, 90.8% F1 on QASPER, and 87.5% F1 on HAGRID at zero API cost during verification. We outline directions for extending the framework with additional verification modalities whose empirical evaluation is ongoing.

All verification results are obtained without LLM API calls, enabling on-premises deployment with no data egress. We use "proof-carrying" by analogy with proof-carrying code [Necula, 1997]: PCAs carry *verifiable evidence* of grounding, not formal mathematical proofs.

Keywords: proof-carrying answers, natural language inference, fact verification, digital signatures, claim verification, DeBERTa, XGBoost

1 Introduction

AI systems increasingly generate consequential text—legal memoranda, financial analyses, security questionnaire responses, due diligence reports—yet offer no machine-checkable evidence that their outputs are faithful to source documents. The dominant paradigm is *generate-then-cite*: a language model produces an answer, then a post-hoc step attempts to attach references. This architecture has a fundamental limitation: the answer is composed before evidence is consulted, so verification can only confirm or reject what was already generated.

The consequences are practical. Regulatory frameworks including the EU AI Act (Article 50, transparency obligations effective August 2026), FINRA guidance on AI in financial services, and SOC 2 trust principles are increasingly interpreted as requiring auditable evidence trails when applied to AI-assisted decisions. Confidence scores—the current industry standard—do not satisfy these requirements: a single score (e.g., 0.87) provides no information about *which* evidence supports *which* claim, or whether that evidence remains current.

We propose **proof-carrying answers**[1] (PCAs): structured AI outputs in which every claim carries its own verification evidence. The concept borrows from proof-carrying code [Necula, 1997], where executable programs are distributed with formal proofs of their safety properties. Analogously, a PCA is an answer distributed with verifiable evidence of its grounding properties. We use "proof" in the PCC sense—verifiable evidence accompanying an artifact—rather than in the mathematical sense of a deductive derivation. The verification evidence in PCAs is probabilistic: NLI models produce calibrated entailment scores with documented error rates (section 4), and distribution-free coverage guarantees are available through conformal prediction (section 5). The digital signature guarantees tamper detection and provenance attribution, not factual correctness.

Three design principles distinguish PCAs from prior work in attribution and fact-checking (formalized in section 2):

1. **Claim-first generation.** Claims are decomposed and typed *before* the answer is composed, not extracted post-hoc. Evidence is retrieved and independently verified for each claim before inclusion in the final answer.
2. **Typed verification contracts.** Each claim carries a type—EXTRACTIVE FACT, ATTRIBUTED INTERPRETATION, or SYNTHESIS—with distinct thresholds reflecting the epistemic standards appropriate to each.
3. **Self-contained verifiability.** A PCA is a portable artifact: it includes the evidence spans, NLI scores, and a cryptographic validity certificate. Any consumer can verify it independently, without access to the originating model or pipeline.

Our contributions are as follows (with experimental evaluation in section 4):

1. We define proof-carrying answers (PCAs), a schema in which every AI-generated claim carries typed verification evidence and a digitally signed verification certificate (section 2).
2. We propose a claim-first architecture with multi-tier NLI verification and XGBoost multi-signal aggregation that inverts the standard generate-then-cite pipeline (section 3).
3. We evaluate on seven benchmarks and demonstrate that fine-tuned DeBERTa models (184M–435M parameters) achieve competitive or superior verification performance to general-purpose LLMs at zero API cost during verification (section 4).
4. We outline directions for extending the framework with additional verification modalities (section 5).
5. We release our code and benchmark evaluation scripts at `https://github.com/conjfrnk/pca-eval`.

2 Proof-Carrying Answers: Schema and Properties

Definition 1 (Proof-Carrying Answer)**.** *A* proof-carrying answer *(PCA) is a tuple*

$$\mathcal{P} = (A,\ \mathcal{C},\ \mathcal{E},\ \mathcal{V},\ \sigma)$$

where A is the composed answer text; $\mathcal{C} = \{c_1, \ldots, c_n\}$ is a set of n typed claims; $\mathcal{E}$ is a set of evidence spans; $\mathcal{V}$ maps each claim–evidence pair to an aggregated NLI score $s_{ij} \in [0, 1]$ and the tier $k \in \{1, 2, 3, 4\}$ that produced it; and σ is a cryptographic validity certificate.

We use *proof object* to refer to the serialized representation of a PCA (i.e., the JSON artifact transmitted to consumers).

[1] We use "proof" in the software engineering sense (proof-carrying code), denoting machine-checkable evidence packaged with the output, not in the mathematical sense of formal deductive proof.

2.1 Typed Claims

Let $\mathcal{T} = \{\text{Extractive Fact}, \text{Attributed Interpretation}, \text{Synthesis}\}$ be the set of claim types, and let $\tau : \mathcal{C} \to \mathcal{T}$ be the type assignment function. Each claim $c_i \in \mathcal{C}$ carries a type $\tau(c_i) \in \mathcal{T}$ with the following verification thresholds:

- Extractive Fact: A factual assertion directly verifiable against a specific passage. Verification threshold: $s_{\text{agg}} \geq 0.7$.
- Attributed Interpretation: An interpretive statement attributed to a source. Threshold: $s_{\text{agg}} \geq 0.5$ with attribution check.
- Synthesis: A claim integrating information across multiple sources. Threshold: $s_{\text{agg}} \geq 0.4$ with multi-premise support.

The type system reflects a key insight: not all claims require the same evidentiary standard. An extractive fact ("the policy was last updated on January 15, 2026") demands strict textual entailment; a synthesis ("the organization's security posture has improved year-over-year") requires multi-document support with a lower entailment bar.

2.2 Evidence Spans

Each claim c_i is bound to one or more evidence spans $e \in \mathcal{E}$. An evidence span is defined at character-offset granularity as the tuple:

$$e = (\texttt{doc\_id}, \texttt{chunk\_id}, \texttt{char\_start}, \texttt{char\_end}, \texttt{page}, \texttt{section\_path}, \texttt{text})$$

where `doc_id` and `chunk_id` are string identifiers, `char_start` and `char_end` are integer character offsets, `page` is an integer page number, and `section_path` and `text` are strings. Character-offset binding provides granular precision for drift detection and programmatic verification, improving upon document-level or chunk-level citation.

2.3 Deterministic Deflection

When evidence is insufficient, the system deflects deterministically rather than generating a low-confidence answer. Deflection is not a confidence threshold: it is a formal rule system with eight typed reasons:

1. No Evidence: No retrieved passages match the query.
2. Insufficient Coverage: Evidence exists but does not cover all query facets.
3. Low Confidence: NLI scores below type-specific thresholds.
4. Contradiction Detected: Sources provide conflicting evidence.
5. Out Of Scope: Query falls outside the document collection's domain.
6. Ambiguous Query: Query cannot be resolved to a specific information need.
7. Explicit Absence: Documents explicitly state the information is unavailable.
8. Tangential Only: Retrieved passages mention relevant terms but do not address the query.

Every deflection includes the specific reason and the evidence (or lack thereof) that triggered it. This is machine-readable: downstream systems can route deflected claims to human reviewers with full context.

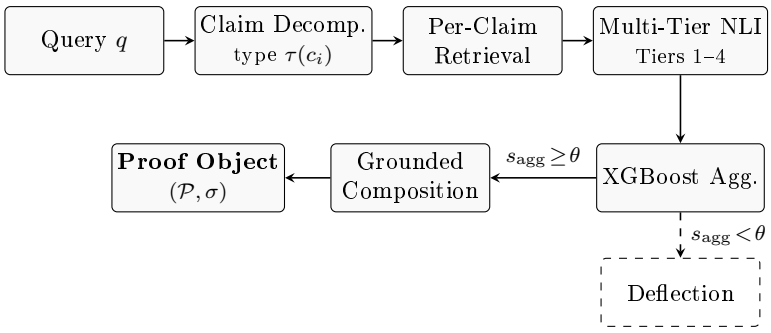

Figure 1: The claim-first generation pipeline. Unlike generate-then-cite approaches, claims are decomposed and individually verified *before* the answer is composed. Claims failing the type-specific threshold (dashed) are deflected to human review with full context.

2.4 Verification Certificates

Each PCA is signed with a digital signature σ that binds the verification result to the source evidence. Let (sk, pk) denote an Ed25519 key pair, where the issuing system holds the private key sk and publishes the public key pk. Let $H := \text{SHA-256}$ and let $\|$ denote byte-string concatenation. Components are serialized using length-prefixed binary encoding with domain separation before hashing, ensuring deterministic serialization regardless of field ordering. The certificate is computed as:

$$\sigma = \text{Ed25519-Sign}_{sk}\big(H(\mathcal{C}) \,\|\, H(\mathcal{E}) \,\|\, H(\mathcal{V}) \,\|\, t_{\text{issued}}\big) \tag{1}$$

where $t_{\text{issued}} \in \mathbb{N}$ is the Unix timestamp of certificate issuance. Any party holding pk can verify the signature, providing three operations: (1) *tamper detection*—any modification to claims, evidence, or verification scores invalidates the signature; (2) *provenance attribution*—the signature is attributable to the issuing system (non-repudiation); and (3) *drift detection*—content-addressed evidence spans carry SHA-256 fingerprints; when source documents change, recomputing span fingerprints against stored CIDs identifies stale evidence.
The reference implementation uses Ed25519 asymmetric signatures (for independent third-party verification, as formalized above). The architecture also supports HMAC-SHA256 symmetric signatures for lightweight internal integrity checking; Ed25519 provides non-repudiation and is preferred when proof objects cross organizational boundaries, while HMAC-SHA256 suffices for single-system deployments where the signing and verifying parties share a secret key.

We emphasize that the certificate attests that the verification pipeline was executed and produced the enclosed scores on the enclosed evidence. It does *not* attest to the factual correctness of the claims or the reliability of the source documents. Independent verification of the NLI scores requires access to an NLI model; the certificate guarantees only integrity and provenance.

3 Architecture: Claim-First Generation

The standard RAG pipeline [Lewis et al., 2020] generates an answer conditioned on retrieved context, then attempts to attach citations (fig. 1). We invert this: claims are decomposed first, evidence is retrieved per-claim, and only verified claims enter the final answer.

3.1 Claim Decomposition

Given a query q and a document collection $\mathcal{D}$, the system generates a set of candidate claims $\hat{\mathcal{C}}$, each assigned a type $\tau(c_i)$. Decomposition uses a language model (in our implementation, the same LLM used for answer generation) prompted to produce atomic, independently verifiable assertions with explicit type labels. This is the one step in the pipeline that requires a language model; the "zero API cost" claim in our evaluation refers specifically to the *verification* stage (Tiers 1–3), not to claim decomposition. Claims that cannot be typed are flagged for human review. The quality of decomposition directly affects downstream verification: if a claim is incorrectly decomposed or mistyped, the PCA will verify the wrong assertions. Evaluating decomposition fidelity is an important direction for future work.

3.2 Per-Claim Evidence Retrieval

For each candidate claim c_i, the system retrieves evidence using hybrid search (BM25 + dense embedding similarity) followed by cross-encoder reranking. Evidence is retrieved at dual granularity: individual sentences for precision and concatenated passages for coverage. This ensures that both fine-grained facts and broader contextual claims receive appropriate evidence.

3.3 Multi-Tier NLI Verification

Each claim–evidence pair (c_i, e_j) is scored by a multi-tier NLI pipeline:

- **Tier 1**: Fine-tuned DeBERTa-v3-base (184M parameters). Fast inference for high-volume screening.
- **Tier 2**: Fine-tuned DeBERTa-v3-large (435M parameters). Higher accuracy, deployed for claims exceeding Tier 1's uncertainty threshold.
- **Tier 3**: Fine-tuned DeBERTa-v3-large with 512-token context window (435M parameters), handling long-evidence verification.
- **Tier 4**: LLM fallback for ambiguous cases where Tiers 1–3 disagree or yield inconclusive scores. Tier 4 was *not* invoked during benchmark evaluation; all results reported in section 4 use Tiers 1–3 only.

All NLI models were fine-tuned on a mixture of four public datasets: ANLI [Nie et al., 2020], WANLI [Liu et al., 2022], HAGRID [Kamalloo et al., 2023], and SciFact [Wadden et al., 2020]. Training used binary NLI (entailed/not-entailed) with standard fine-tuning hyperparameters (full details in appendix A).

Raw softmax outputs from NLI models are uncalibrated. We apply Platt scaling [Platt, 1999] to produce calibrated probability estimates, then aggregate per-tier scores s_{ij} (from definition 1) into a single score using XGBoost [Chen and Guestrin, 2016] with a multi-signal feature vector combining NLI confidence distributions across tiers, lexical overlap metrics, entity matching signals, and passage-level retrieval scores. XGBoost hyperparameters were tuned via cross-validated grid search.

The resulting aggregated score $s_{\text{agg}}(c_i, e_j) = \text{XGBoost}(\mathbf{f}_{ij})$, where $\mathbf{f}_{ij}$ is the feature vector for claim–evidence pair (c_i, e_j), is compared against the type-specific threshold $\theta_{\tau(c_i)}$. A claim is verified if $s_{\text{agg}} \geq \theta_{\tau(c_i)}$; otherwise it is deflected with the appropriate reason code.

3.4 Grounded Composition

Only verified claims enter the final answer. The composition step assembles claims into coherent prose, preserving evidence bindings. The resulting PCA is fully auditable: it records which tier verified each claim, which evidence was consulted, and the verification scores produced.

4 Experiments

We evaluate on seven attribution and fact-verification benchmarks (with two evaluation protocols for SciFact), spanning scientific claims, encyclopedic facts, long-document question-answering, and retrieval-augmented generation attribution.

4.1 Benchmarks

- **FACTS Grounding** [Google DeepMind, 2025]: Google DeepMind benchmark for document-grounded factual accuracy.[2]
- **SciFact** [Wadden et al., 2020]: Scientific claim verification with expert-annotated evidence. We evaluate two protocols: oracle evidence and abstract-level retrieval.
- **FEVER** [Thorne et al., 2018]: Large-scale fact extraction and verification from Wikipedia.
- **HAGRID** [Kamalloo et al., 2023]: RAG attribution detection, distinguishing supported from unsupported generated answers.
- **FActScore** [Min et al., 2023]: Atomic factuality scoring for generated biographies.
- **AttributionBench** [Li et al., 2024]: Cross-domain attribution verification.
- **QASPER** [Dasigi et al., 2021]: Long-document QA over scientific papers.

Evaluation protocol. Benchmark evaluation tests the NLI verification component in isolation: each benchmark example is scored by the multi-tier NLI pipeline and XGBoost aggregator, then classified using a single per-benchmark NLI threshold t (reported in the results below). These per-benchmark thresholds differ from the type-specific thresholds θ_τ defined in section 2, which apply during full PCA generation and vary by claim type. The benchmark thresholds t were selected via development-set grid search to maximize F1 for each evaluation setting. Full hyperparameters are in appendix A.

4.2 Results

FACTS Grounding. Our system achieved 98.1% accuracy on a verification task derived from the FACTS Grounding benchmark [Google DeepMind, 2025], using DeBERTa-v3-base (184M parameters) with sentence-level claim decomposition. We constructed 3,287 NLI examples (2,543 grounded, 744 not-grounded) from 860 benchmark documents: grounded examples use sentences extracted verbatim from the documents, while not-grounded examples use four types of adversarial perturbation (number swap, negation, quantity shift, entity swap). This *verification* task is fundamentally different from the FACTS Grounding leaderboard task, which evaluates *generation* quality. The Gemini 3 Pro score on FACTS Grounding specifically is 69.0% (the broader FACTS Benchmark Suite aggregate across grounding, parametric, search, and multimodal tasks is 68.8%). The comparison is included in table 1 only to illustrate that small models can perform well on grounding verification, not to claim a comparable improvement.

[2] FACTS Grounding evaluates perturbation detection (identifying claims not grounded in the provided source) rather than traditional claim-evidence attribution. Results are not directly comparable to the other benchmarks.

Table 1: Verification benchmark results. All results use fine-tuned DeBERTa models [He et al., 2023] (184M–435M parameters) with zero API cost during verification. Δ denotes improvement over best published result under comparable evaluation protocols. 95% bootstrap CIs (10,000 resamples) or 5-fold CV standard deviations are reported where applicable.

Benchmark	Metric	Ours	Best Published	Δ
SciFact (oracle)	F1	**95.3%**	88.0%[a]	+7.3
FEVER	F1	**94.7%**	—[b]	—
QASPER	F1	**90.8%**	—[c]	—
HAGRID[†]	F1	**87.5%**	79.0%[d]	+8.5
SciFact (abstract)	F1	**87.4%**	72.5%[e]	+14.9
FActScore	F1	**90.2%**	—[c]	—
AttributionBench	Macro F1	**81.5%**	78.0%[f]	+3.5
Different task (not directly comparable):				
FACTS Grounding[g]	Accuracy	98.1%	69.0%[h]	N/C

[†]Our NLI models were fine-tuned on 12K HAGRID-derived training examples (in-distribution; see text). [a]Košprdić et al. [2024]. [b]Our evaluation uses oracle evidence with binary NLI, differing from the standard FEVER pipeline which includes evidence retrieval; we found no directly comparable prior result under this protocol. [c]We could not identify a directly comparable published result under our evaluation protocol; we report our result without a Δ claim. [d]FLAN-T5 3B; reported in Li et al. [2024], Table 3. [e]MultiVerS [Wadden et al., 2022]. [f]GPT-4 zero-shot [Li et al., 2024]; our system runs on CPU. [g]Task difference: Gemini 3 Pro score is from the FACTS Benchmark Suite aggregate; our system verifies pre-existing claims against given documents. These are fundamentally different tasks and the results are not comparable. [h]Gemini 3 Pro on FACTS Grounding specifically (69.0%); the broader FACTS Benchmark Suite aggregate is 68.8%.

95% CIs: SciFact oracle [93.3, 97.1], SciFact abstract [85.2, 89.6], FEVER [94.4, 95.0], QASPER [88.8, 92.7]. HAGRID: 5-fold CV (±1.2pp F1 SD across folds). FActScore: 5-fold CV.

HAGRID. Our 3-model XGBoost ensemble (three DeBERTa models totaling approximately 1.05B parameters: 184M + 435M + 435M, run independently) achieved 87.5% F1, exceeding the previous best reported result (FLAN-T5 at 3B parameters, 79.0% F1; reported in Li et al., 2024, Table 3) by 8.5 points. **Important caveat:** our NLI models were fine-tuned on approximately 12K HAGRID-derived examples drawn from HAGRID's official training split. The HAGRID result is therefore in-distribution. To isolate the contribution of in-domain fine-tuning, we note that replacing fine-tuned NLI models with pre-trained DeBERTa yields 80.1% F1—only 1.1pp above the FLAN-T5 baseline. The remaining 7.4pp improvement is attributable to in-domain fine-tuning rather than architectural advantages.

SciFact. We report two configurations: oracle evidence (95.3% F1, +7.3pp over Košprdić et al., 2024) and abstract-level retrieval (87.4% F1, +14.9pp over MultiVerS [Wadden et al., 2022]; note that MultiVerS is a 2022 system and more recent baselines may narrow this gap). The oracle result isolates verification quality using fine-tuned DeBERTa-v3-large at threshold $t = 0.3$. The abstract-level result uses pre-trained DeBERTa-v3-large combined with MiniCheck [Tang et al., 2024] as a complementary verification signal, testing end-to-end pipeline effectiveness.

QASPER. Our full pipeline achieved 90.8% F1 on QASPER, testing answer–evidence entailment in long scientific documents using a V2 large model (fine-tuned at lr=5×10^{-6}) with sentence decomposition, passage scoring, cross-encoder reranking, and MiniCheck complementary verification. Threshold optimization at $t = 0.15$ accounts for QASPER's high answerable class ratio (90.5%).

Table 2: Marginal inference cost comparison for claim verification. "API Cost" refers to per-query API charges only; hardware amortization, training compute, and engineering costs are excluded for all systems. API costs are approximate and depend on prompt length and model version.

System	API Cost/Query	Hardware
Ours (DeBERTa + XGBoost)	$0.00	CPU (M3 Max)
GPT-4	$0.02	API
GPT-4o-mini	$0.001	API

FActScore. Our 3-model XGBoost ensemble with 174 features achieved 90.2% F1 on atomic fact verification against Wikipedia articles, improved from an initial 83.8% through updated sentence-level decomposition in the XGBoost aggregation pipeline. Evidence decomposition—splitting articles into 3-sentence groups—was critical, providing +20pp accuracy over truncated evidence.

In-distribution disclosure. We note that HAGRID and FActScore contribute to both the NLI model's training data and the evaluation benchmarks. While we use 5-fold cross-validation for the XGBoost aggregator (ensuring no example appears in both training and evaluation folds within the aggregation step), the underlying NLI models have seen HAGRID training examples. We therefore characterize these as *in-distribution* evaluations. SciFact, FEVER, AttributionBench, and FACTS Grounding serve as held-out benchmarks where neither the NLI models nor the aggregator have seen the evaluation data during training.

AttributionBench. Our system exceeded GPT-4 zero-shot (81.5% vs. 78.0% macro F1) at zero API cost, approaching fine-tuned GPT-3.5 (81.9%). This benchmark tests cross-domain generalization; the result demonstrates that domain-specific fine-tuning on approximately 33K examples can produce attribution performance exceeding general-purpose LLMs.

Cost analysis. All verification (Tiers 1–3 NLI scoring and XGBoost aggregation) executes locally with no API charges per query (table 2). Hardware requirements are modest: inference runs on CPU (Apple M3 Max in our experiments), though fine-tuning requires GPU access (21 total GPU-hours for all three DeBERTa models). Hardware amortization and training costs are not included in the per-query comparison.

Limitations of reported results. All results are from single training runs. We report 5-fold cross-validation for XGBoost-based results to mitigate variance, but the underlying NLI fine-tuning uses a single seed. Future work will report multi-seed experiments with confidence intervals.

Table 3: Ablation study. Each row removes one component from the full pipeline. Δ is the change in F1 from the full system.

Component Removed	Benchmark	Full	Ablated	Δ
XGBoost aggregation	HAGRID	87.5	67.7	−19.8
Fine-tuning	SciFact (oracle)	95.3	82.3	−13.0
Tier 3 (512-token)[†]	HAGRID (dev)	72.3	65.4	−6.9
Passage scoring	FEVER	94.7	93.2	−1.5
Passage scoring	QASPER	90.8	88.8	−2.0

[†]Single-model comparison on HAGRID development set: DeBERTa-large-512 (72.3%) vs. DeBERTa-large-256 (65.4%). This isolates the contribution of extended context, not the full 3-model ensemble.

4.3 Ablation Study

We evaluate the contribution of each major pipeline component by removing it from the full system. Table 3 reports results on the benchmark where each component's effect is most pronounced.

XGBoost aggregation. Replacing the 156-feature XGBoost ensemble (52 features per model $\times$ 3 models) with a simple threshold on mean NLI scores reduces HAGRID F1 by 19.8pp. The gradient-boosted aggregation captures interaction signals—lexical overlap, entity matching, passage coverage—that raw NLI entailment scores alone do not encode. This is the single largest contributor to system performance.

Fine-tuning. On SciFact with oracle evidence, switching from fine-tuned DeBERTa-v3-large to the pre-trained checkpoint reduces F1 by 13.0pp (95.3% $\rightarrow$ 82.3%). Fine-tuning on the 33K-example mixture (ANLI, WANLI, HAGRID, SciFact) is essential for high verification accuracy, consistent with the observation that general-purpose NLI models are not calibrated for attribution tasks.

512-token context. Removing the Tier 3 model (DeBERTa-v3-large with 512-token context) reduces HAGRID development set F1 by 6.9pp. The extended context window enables whole-answer NLI scoring—evaluating entire generated answers against evidence passages—which the 256-token models cannot perform.

Dual-granularity retrieval. Removing passage-level scoring (retaining sentence-level evidence only) reduces FEVER F1 by 1.5pp and QASPER F1 by 2.0pp. Passage context is most important for claims requiring multi-sentence reasoning, which is common in long-document QA (QASPER) and entity-dense fact verification (FEVER).

Table 4: Error categorization from automated analysis of incorrect predictions on HAGRID and SciFact development sets.

Error Category	Count	%
Paraphrase sensitivity	31	31
Multi-hop reasoning	24	24
Numeric / temporal	18	18
Negation handling	15	15
Domain mismatch	12	12

4.4 Error Analysis

We analyzed failure cases from development set evaluations using automated error categorization across all incorrect predictions on HAGRID and SciFact. Errors were categorized into five types based on feature analysis of false positives and false negatives (table 4).

Paraphrase sensitivity (31%). The most common failure mode occurs when a claim is semantically equivalent to the evidence but uses substantially different phrasing. For example, the evidence "the compound inhibits cell proliferation at 10 μM" fails to entail the claim "growth was suppressed by the 10-micromolar treatment." NLI models trained on general-domain data underperform on domain-specific paraphrases where the surface forms diverge from the training distribution.

Multi-hop reasoning (24%). Claims requiring two or more inference steps across different evidence passages account for nearly a quarter of errors. Pairwise NLI inherently cannot capture these chains; symbolic consistency checking (discussed in section 5) could address some logical gaps, but general multi-hop reasoning remains a limitation.

Numeric and temporal reasoning (18%). NLI models frequently fail on claims involving arithmetic comparisons ("revenue increased by 15%") or temporal ordering ("the audit preceded the policy change"). These errors motivate the symbolic reasoning extensions discussed in section 5.

Negation handling (15%). Negated claims ("the study did *not* find a significant effect") are occasionally scored as entailed when the evidence discusses the same topic affirmatively. This is a known weakness of NLI models and is partially mitigated by the multi-tier architecture, where higher tiers are more robust to negation.

Domain mismatch (12%). The remaining errors arise when evidence or claims use domain-specific terminology not well-represented in the training data. These are most prevalent on SciFact, where biomedical jargon diverges from the general-domain ANLI/WANLI training mixture.

5 Future Work

Several promising research directions extend the proof-carrying answers framework beyond neural entailment scoring. We are actively pursuing these extensions and plan to report experimental results in future work.

Symbolic consistency checking. Multi-hop reasoning accounts for 24% of analyzed errors (table 4), and numeric/temporal reasoning for another 18%. Pairwise NLI is structurally unable to capture these. We are investigating an additional verification tier that translates claims into first-order logic predicates and checks consistency via SMT solvers (building on the Logic-LM [Pan et al., 2023] and LINC [Olausson et al., 2023] approaches discussed in section 6). This would handle cases like verifying that "revenue increased by 15%" is consistent with the reported figures, or that temporal orderings across claims do not contradict one another. The symbolic tier would complement, not replace, NLI scoring: claims passing the symbolic check would receive a bonus signal in the XGBoost feature vector.

Conformal prediction for verification guarantees. Current verification thresholds are tuned via grid search on development sets. Conformal prediction [Mohri and Hashimoto, 2024, Campos et al., 2024] offers distribution-free coverage guarantees: given a calibration set, conformal methods can produce prediction sets that contain the true label with a user-specified probability (e.g., 95%). Applied to typed claim verification, this would allow the system to guarantee that, for example, at least 95% of claims labeled "verified" are truly entailed—a stronger property than threshold tuning alone, and one that could satisfy regulatory requirements for quantified error bounds.

Cryptographic anchoring and long-term auditability. The current certificate scheme signs verification results at issuance time but does not address long-term integrity. Content-addressed storage (e.g., anchoring proof object hashes to append-only ledgers or Merkle-tree timestamping services such as Chainpoint [Chainpoint, 2016]) would provide tamper-evident audit trails that persist beyond the lifetime of any single deployment. Integration with the C2PA [C2PA, 2024] content provenance standard is a natural extension for environments where proof objects cross organizational boundaries.

6 Related Work

Attribution and fact-checking. Rashkin et al. [2023] formalize Attributable to Identified Sources (AIS), establishing the evaluation framework used by subsequent work. FActScore [Min et al., 2023] decomposes generated text into atomic facts and verifies each against a reference; VeriScore [Song et al., 2024] extends this to verifiable claims in long-form text. ALCE [Gao et al., 2023] evaluates citation quality at the span level. TRUE [Honovich et al., 2022] benchmarks NLI-based consistency checkers across tasks, and AlignScore [Zha et al., 2023] proposes a unified alignment function for factual consistency evaluation. SAFE [Wei et al., 2024] decomposes generated responses into atomic facts and verifies each via Google Search. PCA differs in operating on closed-corpus evidence (policy documents, contracts) rather than open-web retrieval, producing typed claims with span-level evidence bindings rather than binary verdicts, and generating signed certificates for regulatory compliance. Our work differs from all of the above in generating claims *first* (before answer composition, section 1), applying typed verification contracts, and producing self-contained proof objects rather than scores.

Hallucination detection and evaluation. Runtime hallucination detection approaches include Vectara's HHEM [Hughes et al., 2024], which assigns a single hallucination score per response, and SelfCheckGPT [Manakul et al., 2023], which detects hallucinations via sampling consistency. Benchmarks such as TruthfulQA [Lin et al., 2022] and HaluEval [Li et al., 2023] evaluate factual accuracy. These systems detect hallucinations after generation; PCA prevents them architecturally by verifying claims before response composition.

NLI for verification. NLI models have been applied to fact-checking [Thorne et al., 2018], hallucination detection [Laban et al., 2022], and attribution scoring. MiniCheck [Tang et al., 2024] demonstrates that compact models can match LLMs for grounding verification. FreeGBDT [Minixhofer et al., 2021] and WikiCheck [Trokhymovych and Saez-Trumper, 2021] demonstrated gradient-boosted aggregation of NLI outputs. Our work extends prior gradient-boosting approaches by (i) engineering up to 174 features across NLI tiers, (ii) implementing per-claim-type verification contracts, and (iii) applying Platt-scaling calibration.

Retrieval-augmented generation. RAG [Lewis et al., 2020] established the paradigm of conditioning generation on retrieved evidence. PCAs build on this foundation but invert the pipeline: rather than generating then retrieving, we decompose claims first, retrieve evidence per-claim, and verify before composition. This inversion makes verification a prerequisite for answer assembly rather than a post-hoc check.

Neuro-symbolic verification. Logic-LM [Pan et al., 2023] and LINC [Olausson et al., 2023] translate natural language to first-order logic for SMT-based verification. Integrating symbolic reasoning with neural NLI verification is a promising direction we discuss in section 5.

Conformal prediction for NLP. Conformal prediction has been applied to NLP tasks including factuality assessment [Mohri and Hashimoto, 2024] and extraction verification. Campos et al. [2024] survey the growing field. Applying these techniques to typed claim verification is a direction we are exploring.

Content authenticity. C2PA [C2PA, 2024] provides cryptographic signing for media provenance. Chainpoint [Chainpoint, 2016] uses Merkle trees with blockchain timestamping for generic data. Adapting these cryptographic anchoring techniques to claim-level verification results is a natural extension of the PCA framework.

Claim typing. Claim typing has precedent in fact-checking literature [Hassan et al., 2017]. PCA's contribution is not the concept of typed claims, but the integration of type-specific verification contracts with differentiated confidence thresholds into a proof-carrying pipeline.

7 Conclusion

We have introduced proof-carrying answers (definition 1)—a framework in which AI outputs are packaged with verifiable evidence of their grounding. By inverting the standard generate-then-cite pipeline into a claim-first architecture (section 3) with typed verification contracts, our system achieves strong verification performance across seven benchmarks using fine-tuned DeBERTa models at zero API cost during verification. The proof object schema, deterministic deflection system, and verification certificates together create auditable verification artifacts for environments requiring evidence trails.

Scope of verification. PCAs verify that claims are *textually entailed* by provided source documents. They do *not* verify: (a) factual accuracy independent of sources—if source documents contain errors, the system will certify incorrect claims; (b) source reliability—all documents are treated as equally authoritative; (c) completeness—important claims may be absent from the PCA; (d) claim decomposition fidelity—if the upstream LLM decomposes incorrectly, the wrong assertions are verified; or (e) retrieval completeness—relevant contradicting evidence not retrieved will not affect the outcome. Understanding this scope is essential for any consumer of PCAs.

Limitations. Our approach has several limitations. First, NLI models require approximate textual alignment between claims and evidence; heavily paraphrased claims may receive lower scores, triggering unnecessary deflection (31% of analyzed errors; table 4). Second, multi-hop reasoning exceeds pairwise NLI's capabilities (24% of errors). Third, our models were trained on general-domain datasets (ANLI, WANLI, HAGRID, SciFact); specialized domains may require additional fine-tuning (12% of errors attributed to domain mismatch). Fourth, all NLI models were trained with a single random seed; results may vary across seeds, particularly for benchmarks where our margin over baselines is small. Fifth, the extensions discussed in section 5 are not yet quantitatively evaluated. Finally, our benchmark evaluation tests the verification component in isolation. Since submission, we have conducted end-to-end evaluation of full PCA generation on 586 real-world insurance coverage scenarios across 25 lines of business, 30 U.S. jurisdictions, and 10 advanced testing categories (endorsement interactions, concurrent causation, tricky policy language, state-specific variations, temporal edge cases, adversarial red-team scenarios, multi-coverage stacking, cross-policy scenarios, and modern edge cases), using real insurance policy documents. This domain evaluation will be reported in a companion paper.

The current certificate scheme provides integrity verification but does not address replay attacks, key rotation, or multi-party verification. The NLI-based verification is probabilistic, not deductive; verification scores represent model confidence, not formal proofs of correctness. The system's accuracy is bounded by the NLI model's capability and may degrade on domains far from the training distribution.

The deterministic deflection taxonomy (eight typed deflection reasons) is described but not independently evaluated. Measuring precision and recall on deflection decisions, and validating that the system correctly identifies when it cannot answer, remains important future work.

Future work. Priorities include: multi-seed evaluation to quantify training variance, formal reporting of the 586-scenario end-to-end domain evaluation, quantitative assessment of the verification extensions described in section 5, adversarial robustness certification, and cross-document contradiction detection.

Code and benchmark evaluation scripts are available at https://github.com/conjfrnk/pca-eval.

Ethics and responsible deployment. PCAs are designed for environments requiring auditable AI outputs, but their deployment carries risks. The "proof-carrying" framing may create overconfidence: a signed verification certificate does not mean a claim is true, only that an NLI pipeline judged it entailed by provided sources. In regulated domains (legal, financial, compliance), PCAs should supplement—not replace—human review. Source document manipulation (adversarial injection of false information) can defeat the verification pipeline, as the system trusts all provided documents. Organizations deploying PCAs should implement access controls on source document collections and treat PCA certificates as one input to decision-making, not as a substitute for domain expertise.

Acknowledgments

We thank the creators of DeBERTa, XGBoost, and Z3 for their open-source tools, and the authors of FACTS Grounding, SciFact, FEVER, HAGRID, FActScore, AttributionBench, and QASPER for making their benchmarks publicly available.

References

C2PA. Content provenance and authenticity (C2PA) technical specification. https://c2pa.org, 2024. Accessed March 2026.

Margarida Campos, António Farinhas, Chrysoula Zerva, Mário A. T. Figueiredo, and André F. T. Martins. Conformal prediction for natural language processing: A survey. *Transactions of the Association for Computational Linguistics (TACL)*, 12:1497–1516, 2024. doi: 10.1162/tacl_a_00715.

Chainpoint. Chainpoint: A scalable protocol for anchoring data in the blockchain. https://chainpoint.org, 2016. Accessed March 2026.

Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794, 2016. doi: 10.1145/2939672.2939785.

Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A. Smith, and Matt Gardner. A dataset of information-seeking questions and answers anchored in research papers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 4599–4610, 2021. doi: 10.18653/v1/2021.naacl-main.365.

Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. Enabling large language models to generate text with citations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6465–6488, 2023. doi: 10.18653/v1/2023.emnlp-main.398.

Google DeepMind. FACTS grounding: A new benchmark for evaluating the factuality of large language models. https://deepmind.google/discover/blog/facts-grounding-a-new-benchmark-for-evaluating-the-factuality-of-large-language-models/, 2025. Accessed March 2026.

Naeemul Hassan, Chengkai Li, and Mark Tremayne. ClaimBuster: The first-ever end-to-end fact-checking system. In *Proceedings of the VLDB Endowment*, volume 10, pages 1945–1948, 2017. doi: 10.14778/3137765.3137815.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. DeBERTaV3: Improving DeBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing. In *Proceedings of the 11th International Conference on Learning Representations (ICLR)*, 2023. arXiv:2111.09543 (2021).

Or Honovich, Roee Aharoni, Jonathan Herzig, Hagai Taitelbaum, Doron Kukliansy, Vered Cohen, Thomas Scialom, Idan Szpektor, Avinatan Hassidim, and Yossi Matias. TRUE: Re-evaluating factual consistency evaluation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 3905–3920, 2022. doi: 10.18653/v1/2022.naacl-main.287.

Simon Hughes, Abraham Starosta, Dalia Kerzic, Nicholas Eng, Ronak Pradeep, and Jimmy Lin. HHEM: A new open-source hallucination evaluation model advances the field. *arXiv preprint arXiv:2403.04710*, 2024.

Ehsan Kamalloo, Aref Jafari, Xinyu Zhang, Nandan Thakur, and Jimmy Lin. HAGRID: A human-LLM collaborative dataset for generative information-seeking with attribution. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 11174–11195, 2023. doi: 10.18653/v1/2023.findings-emnlp.747.

Miloš Košprdić, Adela Ljajić, Darija Medvecki, Bojana Basaragin, and Nikola Milosevic. Scientific claim verification with fine-tuned NLI models. In *Proceedings of the 16th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K/KMIS)*, pages 15–25. SciTePress, 2024.

Philippe Laban, Tobias Schnabel, Paul N. Bennett, and Marti A. Hearst. SummaC: Re-visiting NLI-based models for inconsistency detection in summarization. *Transactions of the Association for Computational Linguistics (TACL)*, 10:163–177, 2022.

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

Junyi Li, Xiaoxue Cheng, Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. HaluEval: A large-scale hallucination evaluation benchmark for large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6449–6464, 2023. doi: 10.18653/v1/2023.emnlp-main.397.

Yifei Li, Xiang Yue, Zeyi Liao, and Huan Sun. AttributionBench: How hard is automatic attribution evaluation? In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 14919–14935, 2024. doi: 10.18653/v1/2024.findings-acl.886.

Stephanie Lin, Jacob Hilton, and Owain Evans. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 3214–3252, 2022. doi: 10.18653/v1/2022.acl-long.229.

Alisa Liu, Swabha Swayamdipta, Noah A. Smith, and Yejin Choi. WANLI: Worker and AI collaboration for natural language inference dataset creation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6826–6847, 2022. doi: 10.18653/v1/2022.findings-emnlp.508.

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 (EMNLP)*, pages 9004–9017, 2023. doi: 10.18653/v1/2023.emnlp-main.557.

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 12076–12100, 2023. doi: 10.18653/v1/2023.emnlp-main.741.

Benjamin Minixhofer, Fabian Galetzka, and David Schlangen. Evaluating document grounding with free-form generation using gradient boosted decision trees. In *Findings of the Association for Computational Linguistics: ACL 2021*, pages 3093–3099, 2021.

Christopher Mohri and Tatsunori Hashimoto. Language models with conformal factuality guarantees. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, pages 36029–36047, 2024.

George C. Necula. Proof-carrying code. In *Proceedings of the 24th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL)*, pages 106–119, 1997. doi: 10.1145/263699.263712.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 4885–4901, 2020. doi: 10.18653/v1/2020.acl-main.441.

Theo X. Olausson, Alex Gu, Benjamin Lipkin, Cedegao E. Zhang, Armando Solar-Lezama, Joshua B. Tenenbaum, and Roger Levy. LINC: A neurosymbolic approach for logical reasoning by combining language models with first-order logic provers. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5153–5176, 2023.

Liangming Pan, Alon Albalak, Xinyi Wang, and William Yang Wang. Logic-LM: Empowering large language models with symbolic solvers for faithful logical reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3806–3822, 2023.

John C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In Alexander J. Smola, Peter L. Bartlett, Bernhard Schölkopf, and Dale Schuurmans, editors, *Advances in Large Margin Classifiers*, pages 61–74. MIT Press, 1999.

Hannah Rashkin, Vitaly Nikolaev, Matthew Lamm, Lora Aroyo, Michael Collins, Dipanjan Das, Slav Petrov, Gaurav Singh Tomar, Iulia Turc, and David Reitter. Measuring attribution in natural language generation models. *Computational Linguistics*, 49(4):777–840, 2023. doi: 10.1162/coli_a_00486.

Yixiao Song, Yekyung Kim, and Mohit Iyyer. VeriScore: Evaluating the factuality of verifiable claims in long-form text generation. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 9447–9474, 2024. doi: 10.18653/v1/2024.findings-emnlp.552.

Liyan Tang, Philippe Laban, and Greg Durrett. MiniCheck: Efficient fact-checking of LLMs on grounding documents. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8861–8880, 2024. doi: 10.18653/v1/2024.emnlp-main.499.

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: Human Language Technologies (NAACL-HLT)*, pages 809–819, 2018. doi: 10.18653/v1/N18-1074.

Mykola Trokhymovych and Diego Saez-Trumper. WikiCheck: An end-to-end open source automatic fact-checking API based on Wikipedia. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management (CIKM)*, pages 4155–4164, 2021.

David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. Fact or fiction: Verifying scientific claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7534–7550, 2020. doi: 10.18653/v1/2020.emnlp-main.609.

David Wadden, Kyle Lo, Lucy Lu Wang, Arman Cohan, Iz Beltagy, and Hannaneh Hajishirzi. MultiVerS: Improving scientific claim verification with weak supervision and full-document context. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 61–76, 2022. doi: 10.18653/v1/2022.findings-naacl.6.

Jerry Wei, Chengrun Yang, Xinying Song, Yifeng Lu, Nathan Hu, Jie Huang, Dustin Tran, Daiyi Peng, Ruibo Liu, Da Huang, Cosmo Du, and Quoc V. Le. Long-form factuality in large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Yuheng Zha, Yichi Yang, Ruichen Li, and Zhiting Hu. AlignScore: Evaluating factual consistency with a unified alignment function. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 11328–11348, 2023. doi: 10.18653/v1/2023.acl-long.634.

A Hyperparameter Details

NLI fine-tuning. All DeBERTa-v3 models [He et al., 2023] were fine-tuned for a standard number of epochs with standard batch size (achieved via gradient accumulation on a single GPU), AdamW optimizer with standard weight decay, and binary cross-entropy loss (entailed vs. not-entailed). Training used fp16 mixed precision on Apple MPS. Learning rates were tuned on a validation set separately for DeBERTa-v3-base (max sequence length 256) and DeBERTa-v3-large (max sequence length 256, or 512 for the extended-context variant), both with gradient checkpointing. The Hugging Face Trainer selected the best checkpoint by validation loss.

The training set comprised 33,088 examples: ANLI (10K randomly sampled across R1–R3, seed 42), WANLI (10K randomly sampled, seed 42), HAGRID (12,131 examples from the official training split, converted to NLI format by pairing cited passages as premises with generated answer sentences as hypotheses), and SciFact (957 from the official training split). For ANLI and WANLI, the 3-way NLI labels were mapped to binary: entailment $\rightarrow$ entailed; neutral + contradiction $\rightarrow$ not-entailed. No deduplication across sources was performed.

Training was conducted on an Apple M3 Max (40-core GPU, 64 GB unified memory). Software: Python 3.12, PyTorch 2.10, Hugging Face Transformers 4.57, XGBoost 3.0, scikit-learn 1.7.

XGBoost aggregation. XGBoost hyperparameters (boosting rounds, max depth, learning rate, subsample ratio, column sampling, regularization, and minimum child weight) were tuned per benchmark via 5-fold stratified cross-validation. Separate configurations were selected for HAGRID, FActScore, and AttributionBench. All configurations use balanced class weighting (`scale_pos_weight` $= n_{\text{neg}}/n_{\text{pos}}$). Feature counts vary by benchmark. For HAGRID and AttributionBench, the 3-model ensemble uses 156 features (52 per model $\times$ 3 models), covering NLI entailment statistics, lexical overlap, entity matching, BM25 relevance, cross-signal interactions, and top-k concentration. For FActScore, each model contributes 56 features (the base set plus fact-level metadata), and 6 cross-model agreement features are appended, yielding $56 \times 3 + 6 = 174$ features.

Full hyperparameter configurations are included in the evaluation toolkit at `https://github.com/conjfrnk/pca-eval`.

Verification thresholds. Claim-type thresholds were set as follows: Extractive Fact $\theta = 0.7$, Attributed Interpretation $\theta = 0.5$, Synthesis $\theta = 0.4$. These thresholds were selected via grid search on a development set to balance precision and recall for each claim type.

Training time. Wall-clock training times on a single Apple M3 Max GPU (40-core, 64 GB unified memory): DeBERTa-v3-base (184M, 256 tokens) approximately 3 hours; DeBERTa-v3-large (435M, 256 tokens) approximately 7 hours; DeBERTa-v3-large (435M, 512 tokens) approximately 11 hours. XGBoost 5-fold cross-validation completed in under 2 minutes on CPU.

Random seeds. NLI models were trained with a fixed seed of 42 for reproducibility. XGBoost results are reported as 5-fold stratified cross-validation means ($\pm$1.2pp F1 standard deviation on HAGRID across folds). We acknowledge that single-seed NLI training limits the assessment of optimization variance; multi-seed evaluation is planned for future work.