Provenance-Enhanced Retrieval-Augmented Generation with How-Provenance Polynomials

Zubaria Asma[1], Daniel Hernández[2] and Giorgos Flouris[1]

[1]Institute of Computer Science, FORTH, Heraklion, Crete, Greece
[2]University of Stuttgart, Stuttgart, Germany

Abstract

Retrieval-augmented generation (RAG) grounds LLM answers on retrieved evidence, but the answer gives no account of *which* retrieved facts were used or *how* they were combined to produce it. This is a problem beyond mere explainability: even trustworthy sources disagree or contain errors, and a knowledge graph that aggregates sources of unequal quality forces the model to adjudicate between conflicting facts with no visible basis for doing so. We attach *how-provenance polynomials* (semiring provenance) to RAG answers, turning each answer into an auditable expression over its supporting evidence. This serves two purposes: it lets a person inspect which sources an answer rests on, and, more importantly, it gives the *model itself* a structure it can use, together with source-level trust, to identify which of several contradictory retrieved facts is more likely correct. In this position paper we study one facet of this larger problem – whether an uninterpreted provenance signal alone changes how readily an LLM accepts fabricated evidence from untrustworthy sources – and report preliminary results across three open-weight backbones, three prompt phrasings, and sixteen query sets: a bare provenance token, with no explanation of its meaning, cuts acceptance of fabricated facts by roughly 60% relative to an undifferentiated fact list, at a recall cost of only a few points on genuine facts, and outperforms a flat source-lineage baseline carrying the same source information without derivation structure. Combining data from multiple sources to actively identify and resolve contradictory or mistaken facts via substituted trust values is the natural next step and is left, along with several other open questions, to future work.

Keywords

Retrieval-augmented generation, How-provenance, Provenance semirings, Trustworthy LLMs, Knowledge graphs

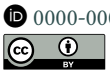
1. Introduction

Retrieval-augmented generation (RAG) grounds an LLM's answer in facts retrieved at query time, and this grounding is usually credited with most of the recent progress against hallucination, or confabulation, the term increasingly used for the same failure. What a RAG answer does not tell you is which of the retrieved items it actually used, or how it combined them: a system that joins two facts to reach a conclusion and a system that picked one fact out of three contradictory ones look identical at the output; a fluent sentence with, at best, a citation appended after the fact. This is adequate when the retrieved evidence agrees with itself, but not when sources of unequal quality are aggregated (e.g., an authoritative registry next to a stale crawl, a curated ontology next to a user-contributed one) because the model then has to adjudicate between conflicting facts with no visible basis for doing so, and neither the model nor the person reading its answer can tell whether that adjudication happened at all.

Provenance semirings [1] were built for exactly this kind of bookkeeping in database query evaluation (Section 2.2); what has not been done is to carry that same polynomial past the retrieval step and into the stage of LLM processing. This will affect the content of the answer that the LLM produces (by taking into account the provenance of the information), but will also ensure that the answer inherits an algebraic account of its own derivation. Carrying the polynomial this far serves two purposes: *explanatory* (a person can inspect exactly which facts and sources an answer rests on), and *corrective* (the same polynomial, evaluated with source-level trust substituted for its variables, gives the *model itself* a basis for choosing between contradictory retrieved facts before it commits to an answer). The

Graph-enhanced LLMs for trustwOrthy Web data management (GLOW), in ISWC 2026 Workshops Joint Proceedings, October 25–26, 2026, Bari, Italy

asma@ics.forth.gr (Z. Asma); daniel.hernandez@ki.uni-stuttgart.de (D. Hernández); fgeo@ics.forth.gr (G. Flouris)
0000-0002-9402-7487 (Z. Asma); 0000-0002-7896-0875 (D. Hernández); 0000-0002-8937-4118 (G. Flouris)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

two are complementary, but it is the second –what the model outputs when its evidence disagrees with itself– that our evaluation (Section 5) is built to measure. This is a position paper: we study one facet of that larger problem, ruling out data from untrustworthy sources using an uninterpreted provenance signal, and leave adjudicating between disagreeing *trustworthy* sources to future work (Section 7).

Concretely, we take the RAG pipeline of Lewis et al. [2] as the setting, annotate retrieved units with provenance variables at the point they enter the pipeline, and combine those variables with the semiring $\cdot$ and $+$ operators as evidence is assembled into a prompt, so each answer carries a checkable record of which facts and sources produced it. Substituting source-level provenance information into the resulting polynomial turns it from a bookkeeping device into a filter: an LLM given both the facts and this structure has, for the first time, something to reason about *before* deciding what to trust, rather than after.

We test whether that structure actually changes model behaviour, not just whether it can be computed. Across 3 open-weight backbones, 3 prompt phrasings, and 16 query sets spanning medical, geographic, and general-knowledge domains, giving a model the same polluted evidence with an uninterpreted provenance token attached (i.e., a provenance record with no explanation of what it means, and with no explicit instruction to filter) cuts acceptance of fabricated facts by roughly 60% relative to the same evidence as a plain list, at a recall cost of a few percentage points (Section 5). In the following, we describe how we build and propagate the polynomial (Section 3), explain the experimental setting (Section 4), report on our experimental results, including a comparison against a flat source-lineage baseline (Section 5), position these results against prior provenance and attributed-RAG work (Section 6), and close with what is still missing before the approach is more than a promising signal (Section 7).

2. Background

2.1. Retrieval-Augmented Generation

Retrieval-augmented generation, as formulated by Lewis et al. [2], pairs a retriever with a generator: the retriever selects passages or facts relevant to a query, and the generator conditions on both the query and the retrieved material to produce an answer. The appeal is that it decouples knowledge from parameters (the generator only needs the ability to use facts it is handed) which is what makes RAG attractive against confabulation, at least when the retrieved evidence is itself correct and uncontested.

The retrieval target does not have to be free text. Knowledge graphs, queried through SPARQL, give a retriever access to structured triples rather than passages, and a growing line of work builds LLM agents directly on top of KG retrieval [3, 4]. This is closer to our setting: a KG triple already carries the subject-predicate-object structure that a provenance-aware layer needs to annotate it.

What none of these systems do is expose the LLM to *how* its context was assembled. Current attribution schemes stop at citation: a retrieved item is tagged with the endpoint or document it came from, but that tag says nothing about whether the answer depended on that source alone, jointly with others, or as one of several redundant alternatives; note that two answers with identical citations can rest on entirely different evidential structures. Closing that gap requires an explicit account of how multiple pieces of evidence, of possibly unequal reliability, were combined to reach the stated answer; this is the province of provenance semirings, which we turn to next.

2.2. How-Provenance and Provenance Semirings

Provenance semirings, due to Green, Karvounarakis and Tannen [1], give database query results an algebraic account of their own derivation. Each base tuple t_i in the input is annotated with a distinct variable x_i from an indeterminate set X; evaluating a query under the semiring $\mathbb{N}[X]$, i.e., polynomials over X with natural-number coefficients, produces for every result tuple a polynomial recording exactly which base tuples contributed and how. A join combines contributing facts multiplicatively: if a result depends on both t_1 and t_2 being true, its provenance is $t_1 \cdot t_2$. A union combines them additively: if the same result follows from either t_1 or t_3 independently, its provenance is $t_1 + t_3$. A result that follows

from t_1 and t_2 jointly, or from t_3 alone, carries the polynomial $t_1 \cdot t_2 + t_3$, which means: "this holds either because of this join, or because of this other fact on its own".

This is *how*-provenance, the strictest member of a family that also includes why- and where-provenance (see [5]): why-provenance (also called *lineage* [5]) records only *whether* a result is possible given the input, discarding multiplicity and combination structure entirely, whereas where-provenance tracks *which* specific source values a result copies from, again without recording how they were combined. Neither distinguishes a fragile, single-witness derivation from a well-corroborated one, a distinction we need once we substitute trust scores for the abstract variables and evaluate the polynomial under an alternative semiring (e.g., $\min/\max$ in place of $\cdot/+$). Everything above is standard database theory with no LLM in the picture; what is new in this paper is not the algebra but where we attach it: carrying the polynomial forward into the context an LLM is given, and asking whether a generator conditioned on that structure behaves differently from one that is not (Section 3).

3. Provenance-Enhanced RAG

3.1. Annotating and Propagating Provenance

Each unit of evidence a retriever returns (a KG triple or text snippet) is annotated at ingestion time with a distinct provenance variable x_i per contributing source; a fact corroborated by more than one source carries one variable per source rather than a single tag, and a fact with no admissible source gets a sentinel "unverifiable" variable. As retrieved units are assembled into the generator's context, these variables are combined using the same $\cdot$ and $+$ semiring operators introduced in Section 2.2: joins multiplicatively, corroborating unions additively. This structure must come from a genuine retrieval process, not be assigned post hoc because two facts merely look similar. We generate disjunctive terms only from queries that return answers found in multiple sources, and conjunctive terms only from compound (join) queries requiring two independent conditions to both hold. The resulting polynomial is attached to the evidence block placed in the prompt, alongside the facts themselves.

3.2. Presenting Provenance in Answers

The polynomial is surfaced to the generator as a compact token column next to each retrieved fact (Section 4.1), with no accompanying explanation of what the notation means; this is a deliberate choice for the evaluation in this paper, since our research question is whether the raw derivation *structure* changes model behaviour on its own, prior to any instruction telling the model what to do with it (Section 5). The same polynomial can equally be surfaced to a person reading the answer, or to a downstream automated check: substituting a qualitative or numeric trust value for each source variable and evaluating the polynomial under an appropriate semiring (e.g., $\min/\max$ for a qualitative lattice such as Schenk's [6]) turns it into a single trust score for the answer as a whole, which either a person or a policy can threshold on. We evaluate only the uninterpreted-token setting in this paper and leave trust substitution to future work (Section 7).

4. Implementation

Code, prompts, query sets, and full per-cell results are publicly available.[1] To test whether a provenance/trust signal changes how an LLM treats retrieved evidence, we built PERAG-Eval, an experimental harness that instantiates the conditions below against live LLM backends, and measures two complementary quantities per query: whether an injected fake fact is *accepted* into the answer, and how much of the *real*, gold-standard evidence is *recalled* (Section 4.4). This way, a filtering effect on the former can be checked against a cost on the latter.

[1] https://github.com/ZubariaForthAcc/PERAGv1

4.1. Domains, Evidence, and Provenance Encoding

The harness spans three domains: *Medical* (drug-drug interactions and contraindications, grounded in a biomedical LOD fact base), *Geographic* (country-level facts from Wikidata/DBpedia/GeoNames/World Bank), and *General Knowledge* (biographical facts). For each domain we retrieve (i) a pool of *real* facts tagged with their contributing knowledge sources (e.g. `wikidata`, `chembl`, `disgenet`), and (ii) a pool of *fake* facts with no verifiable source. Each source is mapped, via a per-domain trust registry, to a provenance token and a qualitative trust label (*High*/*Medium*) following Schenk's [6] qualitative lattice.

Provenance is built compositionally from what the retrieval queries return: a single fact's provenance is the *sum* of its corroborating sources' tokens , degenerating to a single token when the fact is single-sourced, or to the sentinel $t_{unknown}$ when the fact is fake (i.e. has no admissible source), a small number of *join* facts instead take the *product* of two independently-satisfied conditions, and an answer's provenance is the product of its constituent facts' terms. The per-fact annotation shown to the LLM (Section 4.2) is drawn directly from this polynomial rather than assigned after the fact. Disagreeing sources are kept as separate single-witness terms rather than merged into a sum; a normalisation step rules out surface-form differences (e.g. "German" vs. "de") so only genuine value disagreement triggers a split.

Adversarial fake facts. Fakes are constructed as plausible near-misses – off-by-one statistics, a swapped-but-related enzyme, an inflated interaction magnitude – rather than obviously wrong statements, so they cannot be rejected from the LLM's parametric knowledge alone. Pilot runs with implausible fakes collapsed the contrast the experiment measures.

4.2. Experimental Conditions

Five conditions are defined, mirroring Section 5.2: **C1** (parametric only – no retrieved context at all); **C2** (clean RAG – only real facts, no pollution, always the clean baseline regardless of pollution level); **C3** (polluted RAG – real and fake facts as an undifferentiated plain-text list, no provenance); and **C4** (PERAG – the same fact list as C3, but each row is additionally annotated with its provenance token, with injected fakes carrying the sentinel $t_{unknown}$). C4 is run at two pollution levels for the recall study (Section 4.4): **C4@0%** (no fakes injected, isolating any cost of the provenance-table *format* itself, independent of filtering) and **C4@50%** (fakes make up half of the presented facts, exercising filtering under heavy adversarial load). Crucially, C3 and C4 share an identical task instruction and differ *only* in the presence of the provenance column in the data block: no definition of the tokens, and no explicit filtering rule, is given to the model. The LLM must infer, from token structure alone, that a multiply-corroborated sum such as $t_{up} + t_{chembl} + t_{wd}$ signals a well-supported fact while a bare $t_{unknown}$ signals an unverifiable one. This minimal-pairs design isolates the provenance signal's own causal contribution from any instruction-wording confound.

4.3. Prompt Robustness, Backbones, and Execution

Every condition is additionally crossed with three prompt phrasings: *Standard* (a plain list-generation instruction), *Analyst* (a two-step "review, then answer" instruction), and *OPRO-style* (a terse, constraint-first phrasing in the style of optimisation-derived prompts [7]).

The harness talks to LLM backends through a common chat-completion interface (hosted APIs and self-hosted OpenAI-compatible vLLM endpoints). Section 5 uses three open-weight backbones served this way: `gemma-4-26B-A4B-it-FP8-Dynamic`, `llama-3.3-70b-versatile` (via Groq), and `Qwen3.6-35B-A3B-AWQ`, all at a fixed low-temperature decoding setting.

4.4. Response and Recall Scoring

Each response is scored automatically for *fake-fact acceptance*: a binary indicator of whether an injected fake fact appears in the accepted output. For C4, a rejection/omission explanation is first stripped (via hedging patterns such as "the following facts have been omitted") before keyword-matching against a

per-entity signature list, so a fake discussed only in the course of being rejected is not mis-scored as accepted; C1–C3 responses are matched directly.

Every response is additionally scored for *recall* against a hand-curated per-entity gold keyword list, across all five conditions. This lets a recall shortfall in C4@50% be decomposed into a *format* cost (C2 → C4@0%: does the provenance-table format itself suppress recall?) and an *over-rejection* cost (C4@0% → C4@50%: does real filtering cause discarding genuine facts too?). C3 recall is reported for reference only, since it conflates fact retention with pollution acceptance.

4.5. Lineage Baseline: Condition C5

We add a fifth condition, **C5** (lineage-only): the *identical* live-retrieved facts, fake injections, and task prompt as C4, with the sole difference that each fact is annotated with the flat set of sources supporting it (e.g. $\{t_{wd}, t_{dbp}\}$) rather than its full how-provenance term. C4 and C5 are otherwise indistinguishable – same retrieval, same fakes, same wording, same absence of any explanation of what the annotation means – so any difference in outcome is attributable to the presence of derivation structure itself. The C4-vs-C5 comparison (Section 5.5) now covers all three backbones, matching the coverage of Section 5.3: each of `RedHatAI/gemma-4-26B-A4B`, `llama-3.3-70b-versatile`, and `Qwen3.6-35B-A3B-AWQ` was run across all three prompt phrasings and all sixteen query sets, and each comparison was additionally replicated on two further independent runs (Section 5.5).

5. Evaluation

5.1. Research Questions

RQ-E1 Does exposing an LLM to per-fact provenance tokens (Section 4.1), with no explanation of what they mean, reduce its acceptance of fabricated facts relative to the same polluted context without provenance (C3 vs. C4)? **RQ-E2** Is this effect consistent across LLM backbones of different scale and family? **RQ-E3** Is it robust to how the task is phrased, or an artefact of one prompt style? **RQ-E4** Does the size of the effect vary by domain, and if so, which domains remain hardest to defend? **RQ-E5** Does PERAG's fake-rejection come at a measurable cost to the recall of *real* evidence – from the provenance-table format itself, or from over-rejection under heavy pollution? **RQ-E6** Does the algebraic derivation structure of how-provenance (C4) improve fake-fact rejection over a flat lineage set carrying the same source information but no $\cdot/+$ structure (C5)?

5.2. Datasets and Setup

We evaluate on 16 query sets spanning the three domains of Section 4.1: 7 Medical sets grounded in KG-RAG SPOKE [4], MIRAGE [8], and MedHalt [9] (drug interactions, pharmacogenomics, statin safety); 4 Geography sets built on QALD-9 [10], SimpleQA [11], LC-QuAD 2.0 [12], and MultiHal [13]; and 5 General-Knowledge sets built on HotpotQA [14], MuSiQue [15], KGQAGen-10k [16], Defan [17], and a composite General Full Panel drawn from all three.

For each query set we run all three backbones (Section 4.3) under all three prompt phrasings (Section 4.3), scoring fake-fact acceptance (C3, C4) and gold recall (C1, C2, C3, C4@0%, C4@50%). This yields $16 \times 3 \times 3 = 144$ (dataset, model, prompt-style) cells per condition; every cell is filled for both metrics – 432 responses per condition pair, with no missing data, so the results below are not subject to a coverage caveat. Because C1 and C2 contain no injected fakes by construction, the headline fake-acceptance comparison is C3 (polluted RAG, no provenance) vs. C4 (PERAG: the polluted context, with provenance tokens attached); this isolates the marginal value of the provenance signal itself, holding the retrieval content fixed. The recall comparison is C2 vs. C4@0% vs. C4@50%, per Section 4.4.

Data quality. As a sanity check, the per-cell data respects the qualitative ordering the design predicts: C4 > C3 occurs in only 1 of 144 cells and C1 > C2 in only 3, with the remaining residual recall noise (14/144 cells, Section 5.4) consistent with run-to-run variance, which we report rather than exclude.

5.3. Results: Fake-Fact Acceptance

Overall effect (RQ-E1). Averaged across all 144 filled cells, fake-fact acceptance falls from **40.9%** under C3 to **16.1%** under C4 – a 61% relative reduction using only an uninterpreted provenance token, with no explicit filtering instruction.

Table 1
Fake-fact acceptance rate, C3 vs. C4, by domain ($n = 144$; Medical $n = 63$, Geography $n = 36$, General $n = 45$); stable within 0.3pp across two additional replicate runs.

Domain	C3	C4	Rel. reduction
Medical	40.6%	16.5%	59%
Geography	69.0%	28.5%	59%
General Knowledge	18.9%	5.7%	70%

Domain effect (RQ-E4). Table 1 shows the reduction holds in every domain. Geography is hardest to defend in absolute terms (69%→28.5%), plausibly because numeric/demographic figures admit many superficially reasonable values with no obvious internal red flag. General Knowledge has both the lowest baseline (18.9%) and the largest relative gain (70%), the strongest result of the three.

Table 2
Fake-fact acceptance rate, C3 vs. C4, by backbone LLM and by prompt phrasing ($n = 48$ per row); stable within 0.3pp across two additional replicate runs

	C3	C4	Rel. reduction
Gemma-4-26B-A4B	46.0%	16.8%	63%
Llama-3.3-70B	36.5%	14.4%	61%
Qwen3.6-35B	40.3%	17.1%	58%
Standard (list-based)	41.9%	18.8%	55%
Analyst (evaluate + list)	39.1%	19.3%	51%
OPRO-style (optimised)	41.7%	10.3%	75%

Backbone and prompt effect (RQ-E2, RQ-E3). Table 2 shows the C3→C4 lift holds across all three backbones (35B–70B, two families) and all three prompt phrasings, despite no explanation of the tokens' meaning. The relative reduction is stable across backbones (57–63%), consistent with the effect being a property of the signal itself rather than of any one model's instruction-following; across prompts it ranges wider (51–75%), with OPRO-style's terse, constraint-first phrasing giving the strongest result and leaving the most room for the provenance column to do discriminative work. Breaking C4 down by model within domain (not shown for space), Geography remains the weak point for every backbone (26–33% residual acceptance) while General Knowledge is uniformly strong (5–6%).

5.4. Results: Gold Recall and the Cost of Filtering

Overall effect (RQ-E5). Table 3 reports gold-fact recall across all five conditions, averaged over the same 144 cells. Grounding helps recall as expected: C2 (89.2%) is well above C1 (72.7%), confirming that clean retrieval does what it should. More specific to this paper's claim, the *format* cost of PERAG's provenance table is now essentially negligible overall – C4@0% (88.8%) trails C2 (89.2%) by only 0.4 percentage points – while the *over-rejection* cost under heavy pollution is small: C4@50% (86.4%) trails C4@0% by 2.4 points, and trails clean-RAG C2 by 2.8 points. PERAG's roughly 2.5-to-1 reduction in fake-fact acceptance (Section 5.3) therefore comes at a measured cost of under 3 recall points on real evidence, even under adversarial (50%) pollution.

Table 3
Gold recall by condition, overall ($n = 144$ per condition).

	C1	C2	C3[a]	C4@0%	C4@50%
Recall	72.7%	89.2%	86.4%	88.8%	86.4%

[a]C3 recall is shown for reference only and excluded from quality judgements, since C3 also accepts fakes – a high C3 recall number conflates fact retention with pollution acceptance.

Domain, backbone, and prompt effects on recall. Breaking the same figures down (tables omitted for space): Geography carries the largest format *and* over-rejection costs (5.3pp each), consistent with it being hardest to defend (Table 1); Llama has the largest over-rejection gap by backbone (4.3pp); and OPRO, the strongest fake-rejector (Table 2), also has the largest over-rejection gap by prompt (4.2pp) – pushing harder on filtering sweeps out more real evidence too, a trade-off to tune per deployment.

5.5. Results: How-Provenance vs. Simple Lineage

Does the algebra earn its keep? (RQ-E6). Sections 5.3 and 5.4 establish that *some* provenance annotation helps, but not that the specific machinery this paper builds on (semiring polynomials with genuine $\cdot$ and $+$ structure) does more work than a cheaper alternative: tagging each fact with the flat set of sources that support it, and nothing else. Table 4 compares C4 (PERAG) against C5 (lineage-only annotation; Section 4.5) across all three backbones. The polynomial wins consistently, not just on

Table 4
Fake-fact acceptance: C4 (how-provenance) vs. C5 (flat lineage), all three backbones, $n = 432$ (3 replicate runs $\times$ 144 cells).

	C3	C4	C5
Overall	40.9%	**16.2%**	23.9%
Gemma-4-26B-A4B	46.1%	17.0%	25.9%
Llama-3.3-70B	36.4%	14.4%	21.5%
Qwen3.6-35B	40.3%	17.3%	24.5%
Medical	40.6%	16.7%	24.6%
Geography	69.1%	28.5%	40.0%
General Knowledge	18.8%	5.8%	10.2%

average and not just on one backbone: across all $n = 432$ cells (three replicate runs $\times$ 144 cells, spanning all three backbones), C4 outperforms C5 in 402 cells, ties in 24, and loses in only 6. Taken backbone by backbone, C4 beats C5 in 43–48 of every 48 cells for each of the three backbones individually, and the advantage is consistent from run to run. A flat lineage set recovers 39% of the possible reduction in fake-fact acceptance (C3→C5); the full polynomial recovers 60% (C3→C4) – derivation structure captures noticeably more of the effect than source tagging alone, and not at the cost of recall (C4 at 50% pollution, 95.7%, matches or exceeds both clean-RAG C2 and lineage-only C5, on the Gemma backbone). Knowing that a fact has *some* support helps; knowing *how* that support is structured helps further, and this now holds regardless of which backbone is asked.

Limitations. Two limitations bear on these numbers: the provenance signal tested is a bare source token with no natural-language gloss; because injected fakes are annotated with the semantically legible sentinel $t_{unknown}$ while real facts carry source-specific tokens, part of the observed reduction in fake-fact acceptance may reflect the model treating "unknown" as a direct reliability cue rather than exploiting provenance structure per se, and this confound is not disentangled by the current experiment; and the acceptance/recall scorers are keyword-based, so reported rates are automated estimates rather than human-verified ground truth. The headline C4-vs-C5 comparison was replicated on two additional

full runs and now spans all three backbones (Section 4.5); the single-run and single-backbone caveats reported in earlier drafts of this paper no longer apply to the numbers above.

6. Related Work

Provenance in databases and the Semantic Web. How-provenance is not new; what is new is where we attach it. Provenance has a history in the SPARQL and Semantic Web setting, from RDF-entailment derivation tracking to provenance management for KG construction pipelines [18], which tracks how a KG was built rather than how a downstream LLM used a retrieved result. PERAG sits downstream: our NPCS [19] and Fed-NPCS [20] rewriting systems compute how-provenance for the SPARQL queries a retrieval step issues, but this paper's contribution is what happens to that polynomial *after* retrieval.

KG-augmented LLM agents. KG-agent systems use knowledge-graph retrieval to ground LLM reasoning [3], but generally do not expose source heterogeneity or derivation structure to the model. PERAG adds this provenance signal to the retrieved context.

Attribution and provenance-aware RAG. The two closest prior systems attach provenance-like signals to RAG answers directly. SemanticRAG [21] reduces hallucination through answer attribution over KG-verbalised documents; this is shallow attribution rather than derivation-level provenance, because it says which document a claim came from, not whether it depended on that document alone or jointly with others, and thus it carries no notion of source trust (we use it as the attribution baseline C3 in our own evaluation). GenProve [22] goes further, introducing typed, sentence-level provenance at generation time and identifying a reasoning gap around provenance for *inferred* content. Neither propagates a trust score through an algebraic derivation structure; both stop at recording provenance rather than using it to weigh conflicting evidence.

Process provenance and surveys. PROV-AGENT [23] records agent-level process provenance such as prompts, decisions, and tool calls, whereas our polynomials describe how retrieved data supports an answer. Recent surveys [24, 25] place this work within the broader provenance and traceability literature.

Argumentation-based conflict resolution. ArgRAG [26] resolves conflicting evidence using quantitative argumentation rather than algebraic provenance. This is complementary to PERAG: argumentation could consume provenance information, while our approach focuses on an auditable derivation structure.

7. Conclusion and Future Work

We attached how-provenance polynomials to the evidence retrieved by a RAG pipeline and asked a question: does an LLM, given this structure with no explanation of what it means, behave differently from one given the same evidence as a list? Across 3 backbones, 3 prompt phrasings, and 16 query sets, the answer was yes: fake-fact acceptance fell by roughly 60% relative to the no-provenance baseline, at a recall cost on genuine facts of a few points (Section 5). The signal that produced this was deliberately minimal, a bare source-token string with no accompanying gloss, and that minimalism is informative: if an uninterpreted token moves model behaviour this much, the mechanism is giving the model something to notice, arguably a more robust signal than a prompt telling it what to do with the evidence.

That said, the version of PERAG evaluated here is a deliberately narrow slice of the general mechanism described in Section 3, and the gap between the two is the source of future work. Trust propagation was not tested directly: we relied on the raw provenance token doing discriminative work on its own, rather than substituting trust values into the polynomial and evaluating it under an abstract model. Doing so, and testing whether an explanatory gloss helps or dilutes the signal, is the natural next experiment.

Acknowledgments

DH was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC 2120/2 – 390831618.

Declaration on Generative AI

LLMs have been used to produce part of the text in this paper. The authors have carefully reviewed the LLM's output and take full responsibility for the paper's content.

References

[1] T. J. Green, G. Karvounarakis, V. Tannen, Provenance semirings, in: Proc. 26th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS), 2007, pp. 31–40. doi:10.1145/1265530.1265535.

[2] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, S. Riedel, D. Kiela, Retrieval-augmented generation for knowledge-intensive NLP tasks, in: NeurIPS, 2020.

[3] J. Jiang, K. Zhou, W. X. Zhao, Y. Song, C. Zhu, H. Zhu, J.-R. Wen, KG-Agent: An efficient autonomous agent framework for reasoning over KGs, in: ACL 2025, 2025, pp. 9505–9523.

[4] K. Soman, P. W. Rose, J. H. Morris, R. E. Akbas, B. Smith, B. Peetoom, C. Villouta-Reyes, G. Cerono, Y. Shi, A. Rizk-Jackson, S. Israni, C. A. Nelson, S. Huang, S. E. Baranzini, Biomedical knowledge graph-optimized prompt generation for large language models, Bioinformatics 40 (2024) btae560. doi:10.1093/bioinformatics/btae560.

[5] J. Cheney, L. Chiticariu, W.-C. Tan, Provenance in databases: Why, how, and where, Foundations and Trends in Databases 1 (2009) 379–474. doi:10.1561/1900000006.

[6] S. Schenk, On the semantics of trust and caching in the semantic web, in: ISWC, 2008.

[7] C. Yang, X. Wang, Y. Lu, H. Liu, Q. V. Le, D. Zhou, X. Chen, Large language models as optimizers, in: ICLR, 2024.

[8] G. Xiong, Q. Jin, Z. Lu, A. Zhang, Benchmarking retrieval-augmented generation for medicine, in: Findings of ACL 2024, 2024, pp. 6233–6251. doi:10.18653/v1/2024.findings-acl.372.

[9] A. Pal, L. K. Umapathi, M. Sankarasubbu, Med-halt: Medical domain hallucination test for large language models, in: CoNLL, 2023.

[10] R. Usbeck, R. H. Gusmita, et al., 9th challenge on question answering over linked data (QALD-9), in: ISWC Workshops, 2018.

[11] J. Wei, N. Karina, H. W. Chung, Y. J. Jiao, S. Papay, A. Glaese, J. Schulman, W. Fedus, Measuring short-form factuality in large language models, 2024. arXiv:2411.04368.

[12] M. Dubey, D. Banerjee, A. Abdelkawi, J. Lehmann, LC-QuAD 2.0: A large dataset for complex qa over Wikidata and DBpedia, in: ISWC, 2019.

[13] E. Lavrinovics, R. Biswas, K. Hose, J. Bjerva, MultiHal: Multilingual dataset for kg-grounded evaluation of LLM hallucinations, 2025. arXiv:2505.14101.

[14] Z. Yang, P. Qi, S. Zhang, Y. Bengio, W. W. Cohen, R. Salakhutdinov, C. D. Manning, HotpotQA: A dataset for diverse, explainable multi-hop qa, in: EMNLP, 2018.

[15] H. Trivedi, N. Balasubramanian, T. Khot, A. Sabharwal, MuSiQue: Multihop questions via single-hop question composition, TACL 10 (2022) 539–554.

[16] L. Zhang, Z. Jiang, H. Chi, H. Chen, M. Elkoumy, F. Wang, Q. Wu, Z. Zhou, S. Pan, S. Wang, Y. Ma, Diagnosing and addressing pitfalls in KG-RAG datasets, 2025. arXiv:2505.23495.

[17] A. B. M. A. Rahman, S. Anwar, M. Usman, A. Mian, DefAn: Definitive answer dataset for LLMs hallucination evaluation, arXiv preprint arXiv:2406.09155 (2024).

[18] E. Kleinsteuber, T. Al Mustafa, F. Zander, B. König-Ries, S. Babalou, Managing provenance data in knowledge graph management platforms, Datenbank-Spektrum 24 (2024) 43–52. doi:10.1007/s13222-023-00463-0.

[19] Z. Asma, D. Hernández, L. Galárraga, G. Flouris, I. Fundulaki, K. Hose, NPCS: Native provenance computation for SPARQL, in: WWW '24, 2024, pp. 2085–2093. doi:10.1145/3589334.3645557.

[20] Z. Asma, D. Hernández, L. Galárraga, G. Flouris, I. Fundulaki, K. Hose, Native Provenance Computation for Federated and Non-Federated SPARQL Queries, Transactions on Graph Data and

Knowledge 4 (2026) 4:1–4:43. URL: https://drops.dagstuhl.de/entities/document/10.4230/TGDK.4.1.4. doi:10.4230/TGDK.4.1.4.

[21] I. Sapidis, V. Zervos, M. Mountantonakis, Y. Tzitzikas, Interactive and provenance-aware search and QA over documents using LLMs, RAG and KG verbalization, in: TPDL, 2025.

[22] J. Wei, X. Wang, Y. Liao, J. Dong, Y. Liu, C. Jia, B. Yu, J. Zhu, GenProve: Learning to generate text with fine-grained provenance, in: ACL 2026, 2026.

[23] R. Souza, A. Gueroudji, S. DeWitt, D. Rosendo, T. Ghosal, R. Ross, P. Balaprakash, R. F. da Silva, PROV-AGENT: Unified provenance for tracking AI agent interactions in agentic workflows, in: IEEE e-Science, 2025, pp. 467–473. doi:10.1109/eScience65000.2025.00093.

[24] G. Agrawal, T. Kumarage, Z. Alghamdi, H. Liu, Can knowledge graphs reduce hallucinations in LLMs?: A survey, in: NAACL-HLT 2024, 2024.

[25] R. Hohensinner, B. Mutlu, I. G. Mendoza Estrada, M. Vukovic, S. Kopeinik, R. Kern, Tracing the data trail: A survey of data provenance, transparency and traceability in LLMs, JACM 37 (2025) 111.

[26] Y. Zhu, N. Potyka, D. Hernández, Y. He, Z. Ding, B. Xiong, D. Zhou, E. Kharlamov, S. Staab, ArgRAG: Explainable RAG using quantitative bipolar argumentation, in: NeSy, volume 284 of *PMLR*, 2025, pp. 697–718.