RT-2026-06 · PROPOSED

Toward Graph-Conditioned Prefill: Provenance-Scoped Context Assembly for Link Inference

DRAFT An open research track in progress. It sets out a question we are working on inside a live evidentiary environment, reports what has been measured so far, and names what remains untested. Figures will move as experiments run. We are sharing it to find collaborators, not to present a final conclusion.

Evidence: Observed Partner: Systems & KV engineering · experiment ready

Vector retrieval returns what is similar; link inference needs what is reasoning-relevant. We condition context assembly on an analyst-built and adjudicated case graph and report a prerequisite methodological result: when entities are global and relations are source-authored, scoping a subgraph by endpoint membership alone selects the wrong altitude, and scoping it precisely requires per-producer attribution.

Updated August 12, 2026

> **An open research track in progress.** It sets out a question we are working on inside a live evidentiary environment, reports what has been measured so far, and names what remains untested. Figures will move as experiments run. We are sharing it to find collaborators, not to present a final conclusion.

Ask a model to infer a link between two people who appear in different documents and cosine retrieval hands it the passages most similar to the query. Those are rarely the passages that matter. The reasoning-relevant material is often textually dissimilar — a shared address in a filing, a date that lines up with a travel record — and chunking has already destroyed the inter-passage structure the inference depends on. Retrieval optimized for similarity fits a task whose unit of evidence is a relation, not a paragraph, badly.

The argument

An investigative case already has a structure that encodes reasoning relevance: the case graph the analysts built. Entities, relations, adjudications, and corrections are a relevance signal produced by exactly the people whose judgement the model is trying to approximate. Our central hypothesis is that conditioning context on

that graph beats conditioning on embedding similarity, for link inference specifically. It is a hypothesis: the paired comparison that would test it has not been run, and we say so again where it belongs, below.

What runs in production today is graph-conditioned **context assembly**: a layered retrieval stack whose graph tier emits a structured block — nodes, edges, cross-case summary — ordered deterministically ahead of document and annotation chunks, with token budgeting that measures scaffolding cost per run. Both arms of the natural ablation run from a single request parameter, and each is hash-distinguishable, because the assembled context is hashed alongside the template.

We want to be exact about the gap between that and the title. The specified mechanism conditions at the KV level — graph membership deciding which tokens are preserved at full fidelity and which are compressed, backed by a canonical cross-platform cache format. **That mechanism has zero implementation.** A context string and a structured KV cache are different integration points, and every claim we publish must be framed as context assembly until the second one exists.

What we measured

The most useful result so far is methodological, and it is a prerequisite for every number the paired experiment will eventually report.

Entities in our graph are global; relations are authored per document set. That combination makes scoping a subgraph a design question rather than a lookup. The obvious predicate — take every relation whose two endpoints were extracted from this document set — selects the wrong altitude: because the endpoints are shared nodes, it also picks up relations authored in *other* document sets that happen to touch the same entities. Surfacing structure across cases is, for the platform as a whole, the point — it is how the same target seen in two unrelated matters becomes one picture. But inside a *scoped* retrieval experiment it means the graph arm is reading a wider slice than the one under test, and nothing looks anomalous, because the extra context is genuinely relevant — it simply is not the slice you meant to measure.

Defining that scope precisely is **impossible without per-producer attribution**. You cannot tell which document set authored an edge unless every producing model stamped the slice it read. That is now recorded: each producer writes its own entry with a slice identifier, confidence, and span, replacing an earlier collapse of the whole ensemble to a single producer with a null span. A scope predicate can then select edges by where they were actually authored rather than by which entities they happen to touch. The two predicates side by side:

```
naive:   edge.src ∈ scope_nodes  AND  edge.dst ∈ scope_nodes

scoped:  naive
         AND NOT ( edge has ≥1 slice-bearing producer
                   AND every such producer names a document set outside the scope )
         — analyst-authored and unattributed edges are kept
```

The asymmetry is deliberate: an edge is excluded only when another document set is positively shown to have authored it, so an edge with no provenance is kept rather than dropped. For a system built to surface cross-case structure, erring toward inclusion is the right default; it also means how precisely you can scope is a function of how completely provenance has been recorded, not a switch that is simply on or off.

The edges without producer provenance are mostly legacy — written by earlier extractor versions that emitted fewer properties, from before per-producer attribution existed. Scope today is enforced at the document-set (bundle) level; case-level and then system-level scope are deliberate altitudes on the roadmap, not repairs for a defect. The unattributed remainder is enumerable, aging out, and owned.

The practical consequence is narrow: the graph-arm numbers from our earliest offline experiments on the case corpus were reading a wider slice than intended, so they were set aside — an arm is only comparable once both sides are scoped by authorship the same way. We do not assert a direction for that earlier gap; a wider slice is not simply a worse one, only a different question than the one under test.

Link inference is scored against analyst ground truth using accepted-pair recall — predicate-agnostic unordered-pair recall against adjudicated links — alongside adjudicated precision and novel-edge rate, replayed across models on pinned real prompts. A serving artifact matters here too, and it is the kind of thing that quietly ruins a retrieval comparison: a **16k** output cap truncated roughly **29%** of one extractor's valid output, deflating recall for reasons that have nothing to do with what was retrieved.

Corpus and adjudication scale: shared under NDA.

What this does not show

There is no measured flat-versus-graph delta. Both arms are runnable and hash-auditable, the metric is implemented, the graph is provenance-scoped — and no harness runs both arms and diffs them. That missing runner is the critical path, and until it exists the central hypothesis is untested.

The KV-level mechanism is specified and unbuilt. The only prefill latency numbers our infrastructure can produce belong to the serving vendor, and we will not present those as evidence for our mechanism.

Confabulation rate cannot be used to compare the arms. It is computed against the model's own prompt substrate, so adding graph context mechanically widens what counts as grounded. The comparison is confounded by construction; substrate has to be held fixed.

One retrieval path still matches graph nodes without a label, making any latency measured through it a property-scan number rather than an indexed one. And the assistant path runs a deliberately weakened profile, so every number must state which path produced it.

Findings on ungrounded graph closure — the failure mode conditioning should reduce — sit on a typed integrity pool that a recent detector-version change rewrote wholesale. We checked the current pool directly: it is entity-scoped only and carries two classes (fabricated entities, and envelope/attribution artifacts); the graph-closure class this claim would need has zero instances in it, and the rewritten schema has no source-set dimension to concentrate along. We therefore withdraw the concentration claim rather than restate it —

it cannot be re-established on this pool, and any comparison spanning the detector change is invalid. This is a limitation of the current detector's scope, not evidence about the failure mode.

Relationship to prior work

Graph-based retrieval is a crowded and fast-moving area: graph-augmented generation, hybrid textual-relational retrieval, noisy-subgraph filtering, and a growing skeptical literature asking when graph structure earns its cost against plain retrieval at all. "Graphs beat vectors" is not our claim and would not be an interesting one.

The contribution we would defend is narrower and orthogonal to retrieval quality: **provenance-scoped graph retrieval**, and the observation that endpoint membership alone is an insufficient way to scope a subgraph whenever nodes are global and relations are source-authored — it selects by shared entities, not by authorship. The skeptical GraphRAG results connect here in a way that has not been much discussed: a graph arm scoped by the naive predicate is measuring a wider slice than its own, so both positive and negative findings in that setting deserve a second look before they are believed.

- Hu et al., *GRAG: Graph Retrieval-Augmented Generation* — Findings of NAACL 2025 (arXiv:2405.16506)
- *Is GraphRAG Needed? From Basic RAG to Graph-/Agentic Solutions with Context Optimization* — GEM 2026 (arXiv:2606.25656)
- *Awesome-GraphRAG* — survey and benchmark index (DEEP-PolyU/Awesome-GraphRAG)

Where this is going

Write the paired runner: sweep the mode parameter with document input held identical, score accepted-pair recall and adjudicated precision, publish per-arm scorecards. Run the naive-versus-scoped comparison as a worked control on how much the scope altitude moves the numbers. Add a third arm that puts ruled-out relations into context and tests whether telling a model what was excluded suppresses re-emission — the arm for which our track on negative-space supervision supplies the substrate. Then the KV work.

About RedTorch

RedTorch, Inc.® is an investigative intelligence and applied AI firm, operating continuously since 2016 from Inglewood, California under California Bureau of Security and Investigative Services private investigator license 189236. Two practices run in parallel — investigative services (due diligence, litigation support, digital forensics, fraud investigation, asset recovery) and AI and strategy consulting — across more than twenty-five service lines. Casework runs on a platform we built, on GPU infrastructure we own. Privileged material never leaves our custody, and provenance, sanitisation and regulatory mapping are built into the platform's data model and operating workflows rather than added as a reporting layer.

RedTorch's platform is a closed-loop environment for evaluating, improving, and governing models on real evidentiary work. Evidence is captured with its lineage intact; machine claims are checked against their sources; analyst acceptances *and* rejections are preserved as supervision; the case graph rather than

embedding similarity decides what a model reads; competence is measured from production exhaust; and the autonomy a model is granted is gated on that measurement. Each track in this collection documents one stage of that loop. We are building for systems that reason today and act tomorrow — a harder standard than benchmark performance, and a more useful one.

The corpus is what makes the research possible. It combines real privileged investigative text, an industry taxonomy, model-generated claims at scale, and a growing adjudication layer — analyst acceptances, dismissals, ruled-out hypotheses, doctrine assignments and integrity findings. Coverage and attribution vary by signal, and several of those gaps are themselves subjects of these tracks. What cannot be assembled after the fact is the pairing: a decade of privileged casework alongside the machine output produced against it and the expert judgement passed on that output. Corpus metrics are shared under NDA. This track works on the case graphs analysts build inside the platform, which encode a kind of reasoning relevance no embedding recovers.

Collaborate

The unrun experiment we would hand a partner tomorrow is the paired flat-versus-graph ablation at fixed document input and fixed prompt hash, followed by the cross-platform KV divergence curve across quantizations at increasing decode depth. RedTorch, Inc.® supplies analyst-adjudicated link ground truth, both retrieval arms behind one parameter, prompt- and slice-hash auditing, and a provenance-scoped graph. We lack KV-cache engineering capacity and training compute. Researchers and labs interested in this work can get in touch.