Conceptual Evolution Graphs: A Structured Representation of How Ideas Build on One Another

Hita Kambhamettu
University of Pennsylvania
hitakam@seas.upenn.edu

Daniel Krashen
University of Pennsylvania
dkrashen@math.upenn.edu

Lyle Ungar
University of Pennsylvania
ungar@cis.upenn.edu

Abstract

Producing novel hypotheses and arguments requires understanding not just what is known, but how ideas have developed from one another. Existing representations such as knowledge graphs capture factual relations or document citations, but neither encodes how one idea generalizes, refutes, or reformulates another. We introduce the *Conceptual Evolution Graph* (CEG): a graph whose nodes are ideas and whose typed directed edges describe how each idea builds upon prior ones. We construct CEGs using an LLM-based pipeline with two operations: *reduce*, which maps each idea to a canonical form, and *compare*, which draws typed edges between pairs of ideas. We instantiate CEGs on mathematics literature, where ideas are theorems, constructing graphs over 1,213 arXiv papers to model how ideas evolve within mathematical subfields. We then evaluate CEGs on conjecture generation. Across open and closed frontier models, LLM judges match CEG-augmented conjectures to actual published follow-up work at nearly twice the rate of zero-shot prompting (13.8% vs. 8.0%) and substantially more than a citation-graph baseline (9.5%). In a case study, expert mathematicians rated CEG-augmented conjectures as more interesting future directions than baselines (2.5 vs. 1.7 on a 1–5 scale). Our work suggests that representing how ideas evolve, not merely which papers cite which, better supports knowledge-extending tasks.

1 Introduction

Structured representations of knowledge have long played a central role in NLP, both as a source of factual grounding and as a scaffold that makes a model's reasoning steps more interpretable (Luo et al., 2023; Yasunaga et al., 2021; Feng et al., 2020; Yang and Mitchell, 2017; Singhal, 2012). More recently, LLMs have been applied to a qualitatively different class of tasks, which we refer to as knowledge-extending tasks. A knowledge-extending task is one in which a model produces candidate new knowledge, such as claims, ideas, or artifacts not present in an existing knowledge space, by drawing on and reasoning over that space. Representative examples include idea generation, scientific hypothesis formation, and legislative drafting (Wang et al., 2023; Kumar et al., 2024; Li et al., 2024; Hill et al., 2024). New knowledge in these tasks is produced by deeply understanding what is already known and building upon it to produce something new. We argue that supporting such tasks calls for a structured representation of a kind that current ones do not provide: one that encodes not what is known but how it develops, through typed, evolutionary relations between ideas.

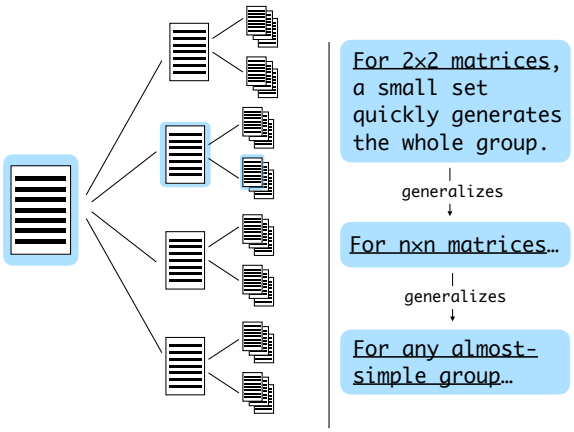

Figure 1: A citation graph (left) versus a Conceptual Evolution Graph (right). While a citation graph shows which papers cite which, a CEG tracks how ideas evolve and build upon others. In this example, a result about 2×2 matrices is generalized first to $n \times n$ matrices and then to a broader class of groups.

Towards this, we introduce the Conceptual Evolution Graph (CEG), a structured knowledge representation in which nodes are ideas and edges are typed evolutionary relations describing how one idea develops from another. We propose an LLM-

based pipeline for constructing CEGs over an existing body of knowledge, organized around two operations: (i) *reduce*, which maps every idea to a shared canonical form, standardizing how each idea is represented so they can be compared directly; and (ii) *compare*, which takes pairs of ideas and assigns typed edges that name how one idea draws from another. The canonical form and schema for assigning typed edges can be changed based on application domain, but the reduce-compare pipeline can be generally applied to construct CEGs (see Section 3.1). Constructing CEGs at this granularity is enabled by LLMs: reading every source document closely enough to extract its central claim and dependencies, and making fine-grained semantic judgments about how pairs of claims relate, is infeasible for human annotators at the scale of an academic subfield, and is beyond what earlier extraction systems could reliably produce. CEGs are thus a structured knowledge representation that is possible because LLMs can perform fine-grained semantic judgments over technical text at scale.

We construct CEGs on academic literature in subfields of mathematics (see Section 3.2). Here, each conceptual idea is a theorem, the canonical form is the dependency graph of the theorem and its supporting lemmas and definitions, and the edges encode how a theorem builds upon others. Constructing CEGs over $1{,}213$ arXiv papers across subfields in mathematics yields a graph that itself admits interesting analysis: we use it to characterize the influence of individual ideas, to trace how lines of work evolve over time, and to identify which evolutionary moves are most prevalent in the field (Section 3.3).

We then evaluate whether CEGs are useful for knowledge-extending tasks. We focus on automated mathematical discovery, an active area of research (Dong and Ma, 2025; Poesia et al., 2024). We hypothesize that this is the kind of task a CEG can support, since fruitful conjectures tend to extend prior work along the same evolutionary modes the CEG discretizes. Concretely, given a starting theorem we ask a model to produce a conjecture that plausibly follows from it, varying the structured representation provided in context. Across both open and closed frontier models (GPT-5.4, Gemini 2.5 Pro, DeepSeek-V3.2, and Claude Opus 4.7), conjectures produced with CEG-augmented prompts are judged by an LLM to be more similar to the actual published follow-up work than those produced with baselines and ablations (averaged across models: 13.8% for CEG vs. 9.5% for a citation-graph baseline, 9.1% for a reduce-only ablation, 8.4% for a compare-only ablation, and 8.0% for zero-shot prompting). In a case study, expert mathematicians evaluated conjectures generated from their own papers and judged CEG-augmented conjectures to be more interesting future directions of the work than baselines (mean 2.5 vs. 1.7 on a 1–5 scale). These results suggest that structured representations of how conceptual ideas evolve can support knowledge-extending tasks.

In sum, we:

- identify conceptual evolution—the *typed, claim-level evolutionary relations* by which one idea develops from another—as a property of structured knowledge representations that is salient for knowledge-extension, the task of producing candidate new knowledge that builds upon existing knowledge;
- introduce the Conceptual Evolution Graph (CEG) and a reduce–compare pipeline for constructing CEGs from an existing body of knowledge, enabled by LLMs' ability to read and compare technical content at a scale beyond what human annotation or earlier extraction methods support, which we instantiate on subfields of mathematics literature;
- demonstrate the utility of CEGs on the knowledge-extending task of mathematical conjecture generation: CEG-augmented LLMs produce conjectures that more closely match actual published follow-up work, and that expert mathematicians judge to be more interesting directions of future work, than citation-graph and ablation baselines.

2 Related work

This section situates our work within two relevant areas: structured knowledge representations that CEGs complement, and the knowledge-extending tasks that CEGs are designed to support.

2.1 Structured knowledge representations

There are many ways to structure knowledge for NLP tasks. Closest to our work are two: knowledge graphs and document graphs. A knowledge graph (KG) models knowledge as a triple (h, r, t), consisting of a head entity, a typed relation, and a tail entity, drawn from a fixed schema (Hogan et al.,

2021). KGs have been used for tasks that require retrieving and reasoning over existing facts, including question answering (Pan et al., 2023; Peng et al., 2023; Saxena et al., 2020), multi-hop reasoning (Feng et al., 2020; Luo et al., 2023), and hallucination detection (Sansford et al., 2024; Chen, 2023; Fang et al., 2024). Temporal KGs extend this representation with timestamps or validity intervals, encoding when a fact holds (Li et al., 2021; Saxena et al., 2021; Cai et al., 2022; Zhu et al., 2020). Work in this line, including methods that model the temporal evolution of TKGs (Li et al., 2021; Zhu et al., 2020), captures how the truth of a relation between fixed entities changes over time. Document graphs structure knowledge at the level of documents rather than individual facts: nodes are documents, and edges link documents that reference one another through citations or hyperlinks. In a document graph over academic papers, for instance, a directed edge from paper A to paper B indicates that A cites B. Prior work uses such networks for related-work recommendation (Cohan et al., 2020), scientific search at corpus scale (Ammar et al., 2018), and document embeddings (Ostendorff et al., 2022).

There has been some work that augments or prunes document graphs: scientific influence graphs quantify that one work shaped another, typically as a weighted or directed link, but leave the nature of that influence unnamed (Wang et al., 2013; Dong et al., 2015). Citation-intent analysis classifies the function a citation serves (Valenzuela et al., 2015a), but operates at the level of paper pairs and labels, encoding why one paper invokes another, not how a particular claim in a paper was used or transformed.

Each of these representations encodes a different aspect of knowledge: KGs factual relations, temporal KGs when those facts hold, document graphs references between documents, citation-intent analysis the function of a citation, influence graphs the existence of influence, and claim extraction the claims themselves. What none provides is the conjunction CEGs are built around: typed relations, defined over individual ideas, that name how one idea develops from others. A KG can record that theorem A was proved by mathematician M but not that A strengthens an earlier theorem B; a citation graph can record that paper X cites paper Y, and citation-intent analysis can determine why, but neither says whether the result in X generalizes, specializes, or refutes a claim in Y. Because CEGs operate at the level of claims within documents, with typed evolutionary edges between them, they encode how knowledge develops.

2.2 Knowledge-extending tasks

We detail what we mean by knowledge-extending tasks in two parts: knowledge, the body of ideas a model reasons over, and extending, the act of producing something new from it. By knowledge we mean conceptual content like theorems, scientific theories, or legal arguments. Unlike an entity, which is a discrete thing the world contains, a piece of knowledge is not self-contained: its meaning depends on prior ideas and on the larger conceptual framework it sits within. By extending we mean producing new content that builds on this existing knowledge in ways that go beyond retrieval and inference, generated instead by reasoning over the structure of what is already known. Tasks in this vein include scientific hypothesis generation (Wang et al., 2023), research idea generation (Si et al., 2025; Jansen, 2026), drug discovery (Maus et al., 2026), and mathematical conjecture generation (Dong and Ma, 2025). Knowledge-extending tasks sit at the intersection of these two: they require both a deep understanding of an existing body of ideas and the generation of something new from it. CEGs uniquely support such tasks because they standardize how each idea is represented and encode the evolutionary relationships through which one idea builds on another.

3 Methods

We now describe how to construct a CEG. We first present the framework in general terms, then describe concrete instantiations that illustrate how the framework's open choices, the canonical form and the edge schema, are made in practice.

The starting point is an existing, connected body of knowledge (such as a citation graph over academic papers) in which concepts are related but the relations do not themselves expose how conceptual ideas evolve. Our goal is twofold: (i) reduce each source of knowledge to a uniform representation that isolates what is evolving, and (ii) compare these representations to identify, and typify, the ways in which one entity has evolved into another. Section 3.1 details the framework; Section 3.2 then describes how we instantiate it in the setting of algebraic geometry literature.

3.1 How to build a Conceptual Evolution Graph (CEG)

The goal of a CEG is to make the evolution of conceptual ideas explicit. We construct one with two operations. The "reduce" operation maps each source document to a canonical form that discretizes the structure of its central claim, so that claims from different sources of knowledge can all be represented in a uniform way. The "compare" operation then takes pairs of reduced concepts in chronological order and draws a typed edge whenever one entity constitutes an evolution of the other one (for example, by strengthening, disproving, or reformulating the earlier claim). Both stages are domain-agnostic; what specializes the pipeline to a domain is the choice of canonical form (in Reduce) and the schema for drawing edges (in Compare). We illustrate each stage on the example in the right side of Figure 1: a lineage in which a result about 2×2 matrices (showing that a small generating set quickly produces the whole group) is generalized to $n \times n$ matrices, and later generalized again to a much broader class of algebraic groups. See Figure 2 for the full pipeline working on this example.

Reduce. The reduce stage maps each source document to a structured representation of its central claim and the dependencies that support it. A shared canonical form is what makes comparison possible: two ideas formulated decades apart, in different notational conventions, are otherwise hard to relate, but if represented in the same standardized form can be compared directly. We prompt an LLM (`gpt-5.1`, `temperature` $= 0.0$) to extract (i) the document's central claim, (ii) the salient supporting concepts that the claim depends on, and (iii) the dependency relations among them. More details in A.6.

In our mathematics instantiation, the central claim is the paper's main theorem, the supporting concepts are its helper lemmas and definitions, and the canonical form is a dependency graph rooted at the main theorem (see the "reduce" section in Figure 2). Applied to our running example, Helfgott (2008) on growth in $SL_2(\mathbb{Z}/p\mathbb{Z})$, the extracted central claim is that every subset A of $SL_2(\mathbb{Z}/p\mathbb{Z})$ either grows rapidly under the group operation or is contained in a proper subgroup. The theorem essentially states that if you take any small set of such matrices A and start multiplying its elements together (allowing inverses), one of two things must happen: either the products you generate quickly fill out the entire group, or your starting set was already trapped inside a smaller substructure.

Compare. The compare stage draws typed edges between concepts that describe how one concept evolves into another. Edges are drawn from a concept that meaningfully supports another one, and each edge type names a specific mode of evolution; together they enable tracking the trajectory of an idea across a body of knowledge. A naive all-pairs comparison is quadratic in the number of concepts and prohibitively expensive at our scale, so we restrict comparisons to likely-related pairs using a two-step retrieval-then-compare procedure. We first embed every concept (with OpenAI's `text-embedding-3-small` (1,536 dimensions)), using the concatenation of the entity's name, formal statement, assumptions, and proof approach. For each entity, we then retrieve all nearest neighbors above a minimum cosine similarity threshold of $\tau = 0.7$. For each of these pairs we prompt an LLM (`gpt-5.1`, `temperature` $= 0$) to judge whether one concept is an evolution of the other, and if so to assign a typed edge from our domain-specific schema. More details in A.7. On our running example, the LLM is presented with the reduced form of Helfgott (2008) (above) and the reduced form of a candidate successor, Breuillard et al. (2011), whose central claim is that the same polylogarithmic-diameter bound holds for $SL_n(\mathbb{Z}/p\mathbb{Z})$ for any n, not only $n = 2$. This is a GENERALIZATION: the conclusion of the predecessor (a small generating set covers the group in $O((\log p)^c)$ steps) is preserved, while the class of groups to which it applies is broadened. The resulting edge is added to the CEG (see "compare" in Figure 2).

Reducing over every source document and comparing over pairs with high-similarity yields a CEG: a directed graph in which paths encode the evolution of conceptual ideas through a knowledge space.

3.2 Building a CEG over algebraic geometry literature

We instantiate the CEG framework of Section 3.1 on academic mathematics literature in subfields of algebraic geometry. We chose this domain as a first application because it offers a particularly clean instantiation of our framework: each paper centers on a theorem, and the ways one theorem can draw upon another (such as generalizing to

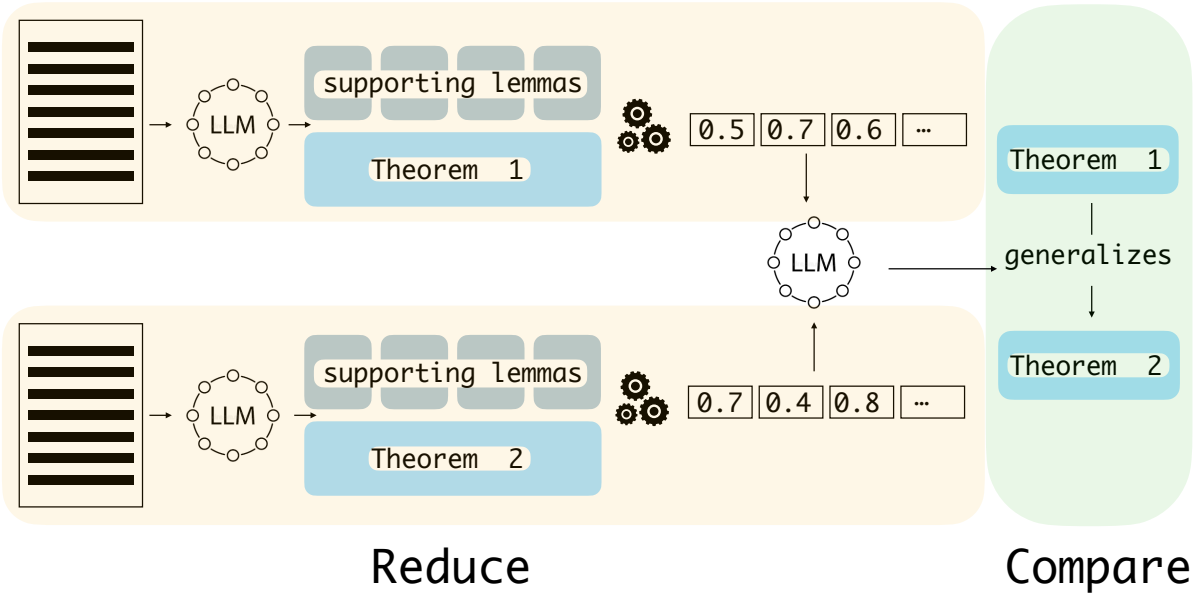

Figure 2: The two-stage CEG construction pipeline. In Reduce (left), an LLM extracts each source document's central theorem and its supporting lemmas, which are then embedded. In Compare (right), we restrict candidate pairs by embedding similarity, then prompt an LLM to judge whether one theorem is an evolution of the other and assign a typed edge (here, generalizes).

new settings or weakening assumptions) result in a small, well-characterized set drawn from prior literature on how mathematics develops (Isaacson, 1978; Pólya, 1990; Kitcher, 1984; Thurston, 2006; Corfield, 2003). In this case, source documents are academic papers, already linked by their references into a citation graph. We use this graph to collect a set of sources on which to create a CEG. Constructing a CEG in this domain takes three steps: collect a set of source papers, reduce each to produce a dependency graph representing the central theorem and the auxiliary lemmas it uses, and compare across pairs of similar ideas. Reduce and Compare follow Section 3.1; what is specific to this setting is the collection step, since the gamut of algebraic-geometry papers is far too large to process exhaustively.

Two collection strategies. We crawl the citation graph in two directions:

- Forward (from a seminal paper outward). Start from an older, influential paper and follow its incoming citations to see how the ideas it introduced were used in later ideas over time.
- Reverse (from recent papers backward). Start from a set of recent papers and follow their outgoing references to see which older ideas they build on.

Both strategies produce a source set on which the same Reduce and Compare pipeline is then run, though they differ in entry point and conceptual direction of the crawl. We analyze the resulting sets separately because each crawl yielded its own self-contained graph, with no substantive evolutionary edges spanning across sets.

Forward collection. In consultation with a domain expert, we identified a seminal paper whose contribution has been actively developed since publication: Bridgeland (2007), which introduces Bridgeland stability conditions and has been cited ∼1,211 times as of writing. Informally, a stability condition is a rule for deciding which mathematical objects in a given category count as "stable" versus "unstable". Bridgeland's contribution was to define such a rule abstractly enough that it applies to a wide range of settings in algebraic geometry, opening the door to many follow-up works. Starting from this seed paper, we use the Semantic Scholar Academic Graph API (Kinney et al., 2023) to retrieve every paper that cites it. For each candidate, we prompt an LLM (GPT-5.1) with the seed paper and the candidate's abstract and ask whether the two are related (prompt in A.8); papers judged related are added to the source set, and this was applied recursively to their own citing papers. To

keep the crawl tractable we cap it at depth 4 or 500 retained papers, whichever comes first. The resulting set contains 500 papers.

Reverse collection. For the reverse direction, we seed from two existing benchmarks of recent arXiv mathematics papers designed for automated scientific discovery tasks: OPENCONJECTURE (Brown, 2026), a dataset of open conjectures extracted from recent arXiv math papers, and ARXIVMATH (Dekoninck et al., 2026), a final-answer benchmark of research problems sourced from recent arXiv papers. We randomly sample 103 seeds from ARXIVMATH and 50 from OPENCONJECTURE, and use the Semantic Scholar API to retrieve the references each seed paper makes. We apply the same LLM-based topical-relatedness filter as in the forward setting, and recurse along references with the same cap of 4 depth or 500 retained papers per seed. The resulting sets contain 427 papers (OPENCONJECTURE) and 286 papers (ARXIVMATH). See A.9 for more details on our instantiation.

3.3 Insights from CEGs

Here, we detail what CEGs reveal about the progression of the ideas in the field we instantiated them in. We provide some evidence that a CEG exposes conceptual structures that go beyond what citation graphs do. We use the three CEGs constructed in Section 3.2 to characterize how ideas evolved in our datasets, both at a higher-level, and at the level of individual chains of evolving theorems.

3.3.1 The distribution of evolution types

The CEGs tell us how conceptual ideas evolve in a given field. Table 1 shows the distribution of edge types in the CEGs constructed. Across all three sets, DIRECT_INVOCATION is the most common cross-paper relation, indicating that most mathematical dependence consists of applying an existing theorem as a step in proving a new one. Perhaps more interestingly, the distributions differ across sets. The BRIDGELAND CEG contains more DOMAIN_EXTENSION and PROOF_TECHNIQUE_TRANSPLANT edges (22.8% and 13.2% respectively, together accounting for 36.0% of all edges). This elucidates the effects of this seminal paper: Bridgeland's stability conditions were introduced as a definition that subsequent work has applied to new triangulated categories, so most descendant work extends the definition's scope rather than modifying its statement. The ARXIVMATH CEG, in contrast, contains more edges that modify theorem statements directly: SPECIALIZATION (11.1%), GENERALIZATION (4.0%), REFORMULATION (3.3%), and CONCLUSION_STRENGTHENING (2.9%) together capture how recent results sharpen or restate prior ones rather than applying them in new domains.

Across all three CEGs, evolution branches broadly from a small number of foundational results rather than chaining deeply. Chains are short — median per-seed length is 1–2 edges in the Bridgeland and ARXIVMATH CEGs, and the longest in either spans 9 edges across 10 papers. Branching, by contrast, is wide at the top: Bridgeland's 2002 paper alone roots 29 immediate successors, but few of those successors themselves root further chains. OPENCONJECTURE is an exception, with both shallow chains (max 2 edges) and minimal branching (average out-degree 0.08): its seeds are recent papers, so little follow-up work has been published yet for the CEG to capture.

3.3.2 Citation vs CEG influence

Citation count has often been used as a proxy for a paper's influence (Tahamtan and Bornmann, 2019; Bertoglio et al., 2021; Raman et al., 2022), but it does not distinguish between reasons for citations (for example, a meaningful extension of work versus an acknowledgment in a background section). Refinements such as Semantic Scholar's "highly influential citations" (Valenzuela et al., 2015b) flag substantive citations, but still operate at the document-citation level and provide only a binary distinction. Because CEG edges connect ideas judged to be in specific evolutionary relationships, a paper's CEG in-degree measures a more targeted notion of influence: how often its central idea has actually been built on or substantively used, and through which mode of evolution. Across all CEGs, the two rankings diverge substantially. The top-10 papers by citation count and the top-10 by CEG in-degree overlap on only 15% of papers on average (mean Spearman $\rho = 0.15$), indicating that CEG in-degree captures a notion of influence that citation count alone does not.

We can also ask which papers are most influential under each kind of evolution: which papers get generalized the most, which papers get applied in new domains the most, and so on for each of the 13 edge types in our CEGs. The answers barely overlap. The most-extended paper is one that does

not lead any other edge type. Different kinds of influence are largely led by different papers. In other words, "influence" in our corpora is not one ranking but many, and they are nearly independent of each other (mean Spearman $\rho = 0.12$ between edge-type rankings). A citation count gives one ranking and surfaces a handful of papers at the top; the CEG gives one ranking per kind of evolution.

4 Experiments

We now evaluate whether CEGs improve LLM performance on a knowledge-extending task. We focus on automated mathematical discovery, a task of growing interest in the AI-for-math community (Dong and Ma, 2025; Poesia et al., 2024). Much of this community has concentrated on proving existing conjectures, often via formalization (Yang et al., 2023; OpenAI, 2026; Norman and Avigad, 2025); the harder (and less-studied) problem is deciding what to prove. A mathematician we consulted estimated that learning to identify fruitful problems takes roughly four years, a skill that is largely tacit and acquired through deep understanding of the literature. We hypothesize that this is exactly the kind of task a CEG can support, since fruitful conjectures tend to extend prior work along the same evolutionary modes the CEG discretizes.

4.1 Setup

Task. We draw a test set of 50 predecessor theorems from the three CEGs constructed in Section 3.2 (randomly sampled 35 from Bridgeland, 10 from OpenConjecture, 5 from ArxivMath). For each predecessor, the model is asked to produce 5 conjectures.

Conditions. We compare five context conditions; prompts for each are in A.10.

- **Zero-shot.** The model sees only the predecessor theorem and relies on its parametric knowledge. This is a no-context baseline.
- **Citation graph.** The model additionally sees the predecessor's incoming and outgoing citations. This is a comparable temporal knowledge-graph baseline, since citation graphs are already used for research-related tasks (Valenzuela et al., 2015b).
- **Reduce-only (ablation).** The model sees the predecessor and the canonical forms of its CEG neighbors, but without edges. This isolates the utility of structured, comparable representations of related theorems.
- **Compare-only (ablation).** The model sees the predecessor together with the typed edge schema and the *names* of edge types incident to it, but without the contents of neighboring theorems. This isolates the contribution of the evolutionary edge information.
- **CEG.** The local neighborhood of the predecessor in the CEG: neighboring canonical forms together with their typed evolutionary edges.

The two ablations are intended to disentangle the contribution of the Reduce and Compare stages: Reduce-only retains the nodes but removes the typed edges, and Compare-only retains the edge schema but removes the node contents.

Models. We run each condition on 4 frontier models spanning open and closed weights: `gpt-5.4` (OpenAI), `gemini-2.5-pro` (Google), `claude-opus-4-7` (Anthropic), and the open-weights `deepseek-v3.2`. All models are run with temperature $= 0.7$ where the API supported it (`gpt-5.4` was run at the provider's default), max output $= 16{,}000$ tokens, and $k = 5$ conjectures sampled per predecessor.

Evaluation. We evaluate each generated conjecture against the actual published follow-up work that cites the predecessor theorem, treating similarity to real follow-up work as a proxy for whether the conjecture is interesting and plausibly true, in line with prior work (Wang et al., 2026). For each predecessor, we retrieve its follow-up papers from the Semantic Scholar citation graph and, for each generated conjecture, ask a judge model whether the conjecture is substantively similar to any of them. We use an ensemble of 3 judge models (`gpt-5.4`, `claude-opus-4-7`, and `gemini-2.5-pro`); a conjecture is counted as a match only if a majority of judges agree.

In pilot experiments, both binary (yes/no) and Likert-scale judgments were noisy. LLM judges either clustered near the midpoint or over-rewarded. We therefore adopt a three-point scale: 3 (substantively similar to a follow-up paper), 2 (related in subject matter but not in substantive content), and 1 (not similar at all). Our primary metric is the 3 rate per condition, computed as the fraction of citing follow-up papers for which at least one of

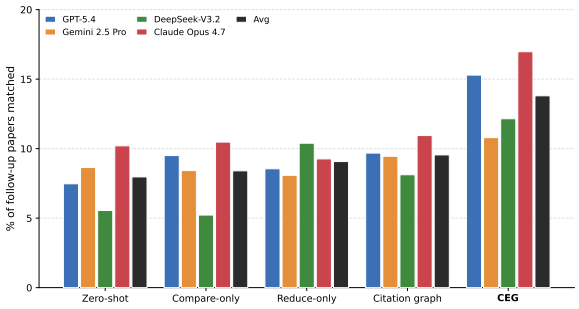

Figure 3: Fraction of citing follow-up papers for which at least one of the $k = 5$ generated conjectures is judged 3 by a majority of judges as being similar to published work for each context condition, broken down by model and averaged across the test set of 50 predecessor theorems.

the $k = 5$ generated conjectures is judged 3 by a majority of judges, averaged across predecessors and models. Full prompt in A.11. Results from an expert validation of the LLM judges in A.12.

4.2 Results

Figure 3 reports, for each generator $\times$ condition cell, the fraction of published follow-up papers for which at least one of the $k = 5$ generated conjectures is judged by the majority of an ensemble of three LLM judges to be substantively similar. CEG-augmented prompting outperforms the baselines (averaged across models: 13.8% for CEG vs. 9.5% for the citation graph, 9.1% for Reduce-only, 8.4% for Compare-only, and 8.0% for zero-shot). A paired bootstrap over predecessors (resampling within each model, pooled across all four generators, n=193 paired observations) places the mean CEG–Citation graph gap at $+4.35$ percentage points with a 95% CI of $[+2.18, +6.56]$, and a Wilcoxon signed-rank test rejects the null of no improvement at $p \approx 1.2 \times 10^{-5}$ (two-sided). The CEG vs. Reduce-only, Compare-only, and Zero-shot contrasts are all significant at $p < 10^{-4}$ (see Appendix A.13 for per-model breakdowns).

4.3 A case study

We also conducted a case study in which two practicing math professors each rated 5 citation-graph-augmented conjectures and 5 CEG-augmented conjectures on 5-point Likert scales for "interesting" (as in, interesting direction for future work) and "plausibly true." Aggregating across the two raters ($n = 10$ per condition), CEG conjectures scored higher than baselines on both axes: 2.50 vs. 1.70 for interestingness and 3.40 vs. 2.89 for plausibility. The gain is more pronounced on interestingness ($+0.80$) than on plausibility ($+0.51$), suggesting that the CEG might help the model surface novel directions of inquiry rather than produce more correct claims. That the expert ratings move in the same direction as the LLM-judged results provides a small but consistent corroboration of the automated evaluation.

5 Discussion

If the reduce–compare pipeline is robust at scale, CEGs offer something other knowledge representations cannot: a typed, claim-level account of how knowledge develops. We see three promising uses. First, surfacing field-aligned candidates for future work: a model that can see how a result has been generalized, strengthened, and reformulated has a better basis for proposing the next move. Second, navigating a literature through evolutionary pathways rather than untyped citation links. A CEG path describes an argumentative trajectory, not just co-occurrence. Third, decomposing influence into many rankings, one per edge type, which Section 3.3 shows diverges from citation count.

We instantiated CEGs on algebraic geometry because the unit of evolution is discrete and finite. Fuzzier domains (legal precedent, clinical guidelines, scientific literature beyond mathematics) will require tailored canonical forms, richer schemas, and careful reliability studies. How the reduce-compare pipeline should be tweaked to yield requisite structure under those conditions is an empirical question we leave open.

6 Limitations

This study has a number of limitations. Our evaluation uses LLM-as-a-judge, with similarity to published follow-up work as a proxy for whether a conjecture is interesting; both introduce error that the expert validation and case study only partially offset. Our pipeline also depends on LLMs at both the Reduce and Compare stages, so extraction errors at either stage propagate downstream. Moreover, our work does not independently validate CEG construction quality at the scale used for evaluation. Finally, our instantiation covers one domain and one unit of evolution (theorems in algebraic geometry); transfer to other settings would require domain-specific choices we do not evaluate here.

7 Ethical Considerations

CEGs might introduce new failure modes once deployed. Mislabeled edges can overstate precedence or novelty, and because the Compare stage retrieves candidate pairs by embedding similarity, recommendations grounded in a CEG could steer researchers toward popular lines of work surfaced by embeddings rather than the genuinely novel directions a CEG is intended to support. Transparent uncertainty estimates, human-in-the-loop validation of high-stakes edges, and provenance visualizations are appropriate near-term mitigations.

8 Conclusion

We argue that knowledge-extending tasks call for a representation organized around evolution, and we introduce the Conceptual Evolution Graph (CEG): a graph where nodes are concepts and edges encode how they build on one another, together with a reduce–compare pipeline for constructing one. Instantiating the pipeline on 1,213 mathematics papers, we show that CEG-augmented in-context learning outperforms a citation-graph baseline and both stage-wise ablations on conjecture generation. CEGs are a first step toward representing knowledge in a way that supports the broader class of knowledge-extending tasks.

References

Waleed Ammar, Dirk Groeneveld, Chandra Bhagavatula, Iz Beltagy, Miles Crawford, Doug Downey, Jason Dunkelberger, Ahmed Elgohary, Sergey Feldman, Vu Ha, Rodney Kinney, Sebastian Kohlmeier, Kyle Lo, Tyler Murray, Hsu-Han Ooi, Matthew Peters, Joanna Power, Sam Skjonsberg, Lucy Lu Wang, and 4 others. 2018. Construction of the literature graph in Semantic Scholar. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), Industry Track*.

R. Bertoglio, C. Corbo, F. Renga, and Matteo Matteucci. 2021. The digital agricultural revolution: A bibliometric analysis literature review. *IEEE Access*, 9:134762–134782.

Emmanuel Breuillard, Ben Green, and Terence Tao. 2011. Approximate subgroups of linear groups. *Geometric and Functional Analysis*, 21(4):774–819.

Tom Bridgeland. 2007. Stability conditions on triangulated categories. *Annals of Mathematics*, 166(2):317–345.

Davis R. Brown. 2026. Openconjecture, a living dataset of mathematics conjectures from the arxiv. `https://github.com/davisrbr/conjectures-arxiv`.

Borui Cai, Yong Xiang, Longxiang Gao, Heng Zhang, Yunfeng Li, and Jianxin Li. 2022. Temporal knowledge graph completion: A survey. In *International Joint Conference on Artificial Intelligence*.

Huajun Chen. 2023. Large knowledge model: Perspectives and challenges. *Data Intell.*, 6:587–620.

Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel S. Weld. 2020. SPECTER: Document-level representation learning using citation-informed transformers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*.

David Corfield. 2003. *Towards a philosophy of real mathematics*. Cambridge University Press.

Jasper Dekoninck, Tim Gehrunger, and Martin Vechev. 2026. Arxivmath: Evaluating llms on mathematical research problems from recent arxiv papers. MathArena Blog. Created by SRI Lab at ETH Zurich and INSAIT.

Kefan Dong and Tengyu Ma. 2025. Stp: Self-play llm theorem provers with iterative conjecturing and proving. *ArXiv*, abs/2502.00212.

Yuxiao Dong, Reid A Johnson, and Nitesh V Chawla. 2015. Will this paper increase your h-index? scientific impact prediction. In *Proceedings of the eighth ACM international conference on web search and data mining*, pages 149–158.

Xinyue Fang, Zhen Huang, Zhiliang Tian, Minghui Fang, Ziyi Pan, Quntian Fang, Zhihua Wen, H. Pan, and Dongsheng Li. 2024. Zero-resource hallucination detection for text generation via graph-based contextual knowledge triples modeling. *ArXiv*, abs/2409.11283.

Yanlin Feng, Xinyue Chen, Bill Yuchen Lin, Peifeng Wang, Jun Yan, and Xiang Ren. 2020. Scalable multi-hop relational reasoning for knowledge-aware question answering. *ArXiv*, abs/2005.00646.

Harald A Helfgott. 2008. Growth and generation in. *Annals of Mathematics*, pages 601–623.

Guzyal Hill, Matthew Waddington, and Leon Qiu. 2024. From pen to algorithm: optimizing legislation for the future with artificial intelligence. *AI & SOCIETY*, 40:3075 – 3086.

Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard de Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, Axel-Cyrille Ngonga Ngomo, Axel Polleres, Sabbir M. Rashid, Anisa Rula, Lukas Schmelzeisen, Juan Sequeda, Steffen Staab, and Antoine Zimmermann. 2021. Knowledge graphs. *ACM Computing Surveys*, 54(4).

Daniel Isaacson. 1978. Proofs and refutations: the logic of mathematical discovery.

Peter A Jansen. 2026. The scientific contribution graph: Automated literature-based technological roadmapping at scale. *arXiv preprint arXiv:2605.15011*.

Rodney Kinney, Chloe Anastasiades, Russell Authur, Iz Beltagy, Jonathan Bragg, Alexandra Buraczynski, Isabel Cachola, Stefan Candra, Yoganand Chandrasekhar, Arman Cohan, and 1 others. 2023. The semantic scholar open data platform. *Preprint*, arXiv:2301.10140.

Philip Kitcher. 1984. *The nature of mathematical knowledge*. Oxford University Press.

Sandeep Kumar, Tirthankar Ghosal, Vinayak Goyal, and Asif Ekbal. 2024. Can large language models unlock novel scientific research ideas? *ArXiv*, abs/2409.06185.

Long Li, Weiwen Xu, Jiayan Guo, Ruochen Zhao, Xinxuan Li, Yuqian Yuan, Boqiang Zhang, Yuming Jiang, Yifei Xin, Ronghao Dang, Deli Zhao, Yu Rong, Tian Feng, and Li Bing. 2024. Chain of ideas: Revolutionizing research via novel idea development with llm agents. *ArXiv*, abs/2410.13185.

Zixuan Li, Xiaolong Jin, Wei Li, Saiping Guan, Jiafeng Guo, Huawei Shen, Yuanzhuo Wang, and Xueqi Cheng. 2021. Temporal knowledge graph reasoning based on evolutional representation learning. *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Linhao Luo, Yuan-Fang Li, Gholamreza Haffari, and Shirui Pan. 2023. Reasoning on graphs: Faithful and interpretable large language model reasoning. *ArXiv*, abs/2310.01061.

Yury A. Malkov and contributors. 2026. hnswlib. Accessed: 2026-05-25.

Natalie Maus, Yimeng Zeng, Haydn Thomas Jones, Yining Huang, Gaurav Ng Goel, Alden Rose, Kyurae Kim, Hyun-Su Lee, Marcelo Der Torossian Torres, Fangping Wan, and 1 others. 2026. Purely agentic black-box optimization for biological design. *arXiv preprint arXiv:2601.22382*.

Chase Norman and Jeremy Avigad. 2025. Canonical for automated theorem proving in lean. In *International Conference on Interactive Theorem Proving*.

OpenAI. 2026. An OpenAI model has disproved a central conjecture in discrete geometry. https://openai.com/index/model-disproves-discrete-geometry-conjecture/. Accessed: 2026-05-25.

Malte Ostendorff, Nils Rethmeier, Isabelle Augenstein, Bela Gipp, and Georg Rehm. 2022. Neighborhood contrastive learning for scientific document representations with citation embeddings. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. 2023. Unifying large language models and knowledge graphs: A roadmap. *IEEE Transactions on Knowledge and Data Engineering*, 36:3580–3599.

Ciyuan Peng, Feng Xia, Mehdi Naseriparsa, and Francesco Osborne. 2023. Knowledge graphs: Opportunities and challenges. *Artificial Intelligence Review*, pages 1 – 32.

Gabriel Poesia, David Broman, Nick Haber, and Noah D. Goodman. 2024. Learning formal mathematics from intrinsic motivation. *ArXiv*, abs/2407.00695.

George Pólya. 1990. *Mathematics and plausible reasoning: Induction and analogy in mathematics*, volume 1. Princeton University Press.

R. Raman, K. Achuthan, V. Nair, and Prema Nedungadi. 2022. Virtual laboratories- a historical review and bibliometric analysis of the past three decades. *Education and Information Technologies*, 27:11055 – 11087.

Hannah Sansford, N. Richardson, Hermina Petric Maretic, and Juba Nait Saada. 2024. Grapheval: A knowledge-graph based llm hallucination evaluation framework. In *KiL@KDD*.

Apoorv Saxena, Soumen Chakrabarti, and Partha P. Talukdar. 2021. Question answering over temporal knowledge graphs. *ArXiv*, abs/2106.01515.

Apoorv Saxena, Aditay Tripathi, and P. Talukdar. 2020. Improving multi-hop question answering over knowledge graphs using knowledge base embeddings. In *Annual Meeting of the Association for Computational Linguistics*.

Chenglei Si, Diyi Yang, and Tatsunori Hashimoto. 2025. Can llms generate novel research ideas? a large-scale human study with 100+ nlp researchers. In *International Conference on Learning Representations*, volume 2025, pages 94003–94092.

Jeremy Singer-Vine. 2026. pdfplumber. Accessed: 2026-05-25.

Amit Singhal. 2012. Introducing the knowledge graph: things, not strings. The Keyword (Google Blog). Accessed: 2026-05-25.

I. Tahamtan and L. Bornmann. 2019. What do citation counts measure? an updated review of studies on citations in scientific documents published between 2006 and 2018. *Scientometrics*, 121:1635 – 1684.

William P Thurston. 2006. On proof and progress in mathematics. In *18 Unconventional essays on the nature of mathematics*, pages 37–55. Springer.

Marco Valenzuela, Vu Ha, and Oren Etzioni. 2015a. Identifying meaningful citations. In *AAAI workshop: Scholarly big data*, volume 15, page 13.

Marco Valenzuela, Vu A. Ha, and Oren Etzioni. 2015b. Identifying meaningful citations. In *AAAI Workshop: Scholarly Big Data*.

Dashun Wang, Chaoming Song, and Albert-László Barabási. 2013. Quantifying long-term scientific impact. *Science*, 342(6154):127–132.

Heng Wang, Pengcheng Jiang, Jiashuo Sun, Zhiyi Shi, Haofei Yu, Jiawei Han, and Heng Ji. 2026. Learning to predict future-aligned research proposals with language models. *arXiv preprint arXiv:2603.27146*.

Qingyun Wang, Doug Downey, Heng Ji, and Tom Hope. 2023. Scimon: Scientific inspiration machines optimized for novelty. In *Annual Meeting of the Association for Computational Linguistics*.

Bishan Yang and Tom Michael Mitchell. 2017. Leveraging knowledge bases in lstms for improving machine reading. In *Annual Meeting of the Association for Computational Linguistics*.

Kaiyu Yang, Aidan Swope, Alex Gu, Rahul Chalamala, Peiyang Song, Shixing Yu, Saad Godil, Ryan J Prenger, and Animashree Anandkumar. 2023. Leandojo: Theorem proving with retrieval-augmented language models. *Advances in Neural Information Processing Systems*, 36:21573–21612.

Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and J. Leskovec. 2021. Qa-gnn: Reasoning with language models and knowledge graphs for question answering. In *North American Chapter of the Association for Computational Linguistics*.

Cunchao Zhu, Muhao Chen, Changjun Fan, Guangquan Cheng, and Yang Zhan. 2020. Learning from history: Modeling temporal knowledge graphs with sequential copy-generation networks. *ArXiv*, abs/2012.08492.

A Appendix

A.1 Use of artifacts

All source data used in this work is publicly available: arXiv papers are distributed under their authors' chosen licenses (typically arXiv's non-exclusive license or various Creative Commons licenses), the Semantic Scholar Academic Graph API is used in accordance with its terms of service, and the OPENCONJECTURE and ARXIVMATH benchmarks are publicly released by their authors. We use commercial LLM APIs (OpenAI, Google, Anthropic) in accordance with their respective terms of service. The CEGs constructed in this work will be released under CC-BY-4.0 for research use. Our use of existing artifacts is consistent with their intended use: arXiv papers are publicly disseminated for scientific use, the Semantic Scholar API is provided for research, and OPENCONJECTURE and ARXIVMATH are research benchmarks. The CEGs we construct are derived from research literature and intended for research use only; they are not appropriate for non-research contexts.

A.2 Personally Identifying Information and Offensive Content

Source data consists of publicly available arXiv mathematics papers; author names appearing in these papers are public attributions on published academic work and not considered personally identifying information in the sense of sensitive personal data. We did not identify offensive content in our source data, which is consistent with the nature of mathematical research papers. For the case study in Section 4.3 involving two mathematicians, individual ratings are reported only in aggregate.

A.3 Models and Computational Budget

We use commercial LLM APIs for all generation and judgment tasks: OpenAI `gpt-5.1` and `gpt-5.4`, Google `gemini-2.5-pro`, Anthropic `claude-opus-4-7`, and the open-weights `deepseek-v3.2`. Parameter counts for the closed models are not publicly disclosed. CEG construction over the 1,213 papers required approximately 1,213 Reduce calls (one per paper, on the full LaTeX source) and approximately 47,300 Compare calls (one per candidate cross-paper pair surfaced by the embedding retrieval), in addition to approximately 6,500 abstract-relatedness filter calls during the crawl. We estimate this consumed on the order of $\sim$ 350M input tokens and $\sim$ 55M output tokens, at an approximate API cost of $\sim$ \$2,500. The conjecture generation experiments consumed approximately 200M tokens across 1,000 generation calls (50 predecessors $\times$ 4 models $\times$ 5 conditions) and 3,000 judge calls (3 judges per generation), at an approximate cost of $\sim$\$1,200. Embedding (OpenAI `text-embedding-3-small`, 1,536 dimensions) and HNSW retrieval ran on a single workstation (M1 Macbook Pro with 32GB RAM).

A.4 Instructions for Human Raters

Two practicing mathematics professors served as raters in the case study described in Section 4.3.

Each was shown a predecessor theorem and 10 candidate conjectures (5 generated with citation-graph context and 5 generated with CEG context, presented in randomized order with condition labels removed). They were asked to rate each conjecture on two 5-point Likert scales:

- **Interestingness** (1–5): "How interesting is this conjecture as a direction for future work in your field?"
- **Plausibility** (1–5): "How plausibly true does this conjecture seem to you?"

Raters were informed that the conjectures were generated by LLMs under different context conditions, that their individual ratings would be aggregated and reported in research output, and that no personally identifying information about them would be shared. Both mathematicians were recruited through the authors' professional contacts at a university mathematics department. They participated voluntarily as part of an ongoing research collaboration and received no monetary compensation.

A.5 Use of AI

We use Generative AI in this project. We use Claude Code to help implement experiments and Claude and Gemini to help refine paper writing.

A.6 Implementation Details of "Reduce"

A.6.1 Pipeline

The reduce stage proceeds in three steps:

1. For each paper, we download the arXiv PDF directly from `https://arxiv.org/pdf/{arxiv_id}.pdf` (using the legacy `/pdf/{category}/{id}` URL for pre-2007 identifiers). All downloads are cached on disk by arXiv id so that re-runs are deterministic.

2. We extract raw text from the PDF with `pdfplumber` (Singer-Vine, 2026). Pages are concatenated with newline separators and the full extracted text is passed to the next step without truncation; we found that truncating to a fixed character budget routinely cut off the main theorem or its supporting lemmas, especially in algebraic geometry papers where the main theorem may appear after 5–10 pages of background.

3. The full extracted text is fed to a single LLM call (`gpt-5.1`, `temperature`=0.0) with the prompt in Figure 4, which asks for (i) the paper's main theorem as a complete LaTeX statement, and (ii) every definition, lemma, proposition, and corollary the main theorem depends on (directly or transitively), each tagged with a list of ids of the other nodes it depends on. The response is parsed as JSON and validated against the schema implied by the prompt (the main theorem must exist, every `depends_on` id must reference a node defined in the same response, and the resulting graph must be acyclic).

A.6.2 Prompt

Figure 4 shows the prompt used. The two interpolated fields, `{paper_title}` and `{paper_text}`, are filled with the arXiv title and the `pdfplumber`-extracted text respectively. We instruct the model to output a full LaTeX statement for the main theorem and one-sentence summaries for all supporting nodes; this asymmetry was chosen so that the main theorem can be re-rendered faithfully in the compare stage while supporting nodes remain compact enough to embed and compare without exhausting context windows.

A.6.3 Some notes on our design choices

- Roughly 20% of our algebraic-geometry corpus has the main theorem stated after page 4. Truncating the LLM input to a fixed token budget caused a measurable drop in extraction success; we therefore pass the entire PDF text and rely on the LLM's long-context handling.
- We do not split the paper into chunks. The DAG structure relies on the model seeing every supporting lemma in the same context as the main theorem; chunking messes with the dependency relations.
- Extraction is deterministic per arXiv id at `temperature`=0.0 up to LLM nondeterminism; cached JSON outputs are reused across all downstream experiments to ensure that compare-stage results are not confounded by extraction noise.

A.7 Implementation Details of "Compare"

A.7.1 Pipeline

The compare stage works in three steps:

1. Every reduced entity (main theorem plus every supporting node from Section A.6) is embedded with OpenAI's `text-embedding-3-small` model (1,536 dimensions). The string that is embedded is the concatenation of the entity's `name`, formal `statement`, `assumptions`, and `proof_sketch` in that order, separated by newlines. We embed full statements because the typed-edge classifier needs access to the mathematical content of both sides of the edge.

2. Embeddings are inserted into an `hnswlib` (Malkov and contributors, 2026) index (`space=l2`, `ef_construction`=200, M=16). For each entity we query the index for its nearest neighbors under L2 distance (equivalent to cosine similarity on the unit-norm OpenAI embeddings), excluding self-matches. To avoid spending LLM calls on obvious non-matches, we discard any pair whose cosine similarity is below $\tau = 0.7$.

3. Every resulting pair is passed to an LLM (`gpt-5.1`, `temperature`=0) with the prompt in Figure 5. The prompt fills in the names, statements, assumptions, proof approaches, and surrounding paper context for both claims and asks the model to (i) decide whether the later claim meaningfully depends on the earlier one and (ii) if so, choose exactly one of the 13 typed-edge categories defined in the prompt (plus a 14th "none" option). The response also includes a `dependency_strength` field (`strong`/`moderate`/`weak`/`none`) and a free-text `explanation` and `evidence` pair so we can audit individual edges. We retain only edges with `has_dependency`=`true` and `dependency_strength`≥`moderate`.

A.7.2 Prompt

Figure 5 has the prompt we used. The fields `{theorem1_*}` and `{theorem2_*}` are filled with the reduced-form fields of the older and newer theorem respectively; `{theorem2_context}` is a short excerpt of the surrounding text (paper title + abstract) of the newer entity's paper, which we found materially improved the model's calibration on `direct_invocation` vs. `analogy`.

A.7.3 Some notes on our design choices

- We chose `text-embedding-3-small` (1,536 dims) because the embedding is only a coarse retriever. The LLM outputs the actual edge type, so the model only needs to keep true neighbors in the top-k, not rank them perfectly. At this role, `3-small` is roughly 6× cheaper per token than `text-embedding-3-large` (3,072 dims) and is sufficient for k-NN over the few hundred to few thousand entities in each CEG.

- We swept $\tau \in \{0.5, 0.6, 0.65, 0.7, 0.75\}$ on a held-out slice of one of our corpora and chose the threshold that maximized agreement between the LLM's `has_dependency`=`true` rate and a hand-annotated ground truth on 5 pairs. Above 0.7 recall drops sharply; below 0.7 the LLM call rate balloons without proportional precision gain.

- We instruct the LLM that Theorem 1 is the candidate predecessor and Theorem 2 is the candidate successor; the typed edge is added with this directionality. When the two entities are from different papers, the predecessor / successor assignment is given by arXiv publication year; ties are broken by submission date.

- Both embedding and pairwise-comparison results are cached on disk by entity-pair id, so the compare stage is deterministic across reruns up to LLM nondeterminism at `temperature`=0 and can be extended with new entities.

A.8 Abstract-Level Filter Prompt

To make the BFS over citations tractable, before any theorem-level comparison we run a lightweight abstract-level filter on each candidate paper–paper edge. Given the title and abstract of both the cited paper (parent) and the citing paper (candidate child), the filter decides whether the citing paper's results plausibly depend on the cited paper's at the theorem level (rather than merely citing it for background or motivation). Only paper–paper pairs that the filter accepts proceed to the reduce stage and theorem-pair comparison. The exact prompt is reproduced in Figure 6; it is called with `gpt-5.4`, `temperature`=0.

Edge type	Description	Frequency
direct_invocation	step in proving new result	67.3%
specialization	narrower case of result	10.0%
domain_extension	applied in new setting	4.6%
generalization	broader case of result	3.5%
reformulation	same result, restated	3.3%
proof_technique_transplant	proof method borrowed	3.1%
abstraction_unification	unified under common frame	2.6%
conclusion_strengthening	conclusion tightened	2.6%
alternative_proof	different proof, same result	1.0%
counterexample_tightness	bound shown tight	0.8%
analogy	structural parallel only	0.5%
assumption_weakening	hypothesis relaxed	0.3%
lemma_incorporation	sub-lemma reused	0.2%

Table 1: Frequency of each edge type across CEGs we constructed in mathematics literature.

A.9 Instantiating Reduce–Compare on Algebraic Geometry Literature

Reduce. For each paper in a source set, we prompt an LLM (`gpt-5.4`) [1] with the paper's full LaTeX source and instruct it to extract the central theorem together with the helper lemmas and definitions on which it depends, returned as a self-contained DAG (prompt in Figure 4). This DAG is the canonical form to represent each entity in this CEG.

Compare. We embed each DAG with an embedding model (OpenAI's `text-embedding-3-small`) using the concatenation of theorem name, statement, assumptions, and proof approach as the textual basis, and retain candidate pairs from the nearest neighbors per node by cosine similarity (HNSW index). For each of these pairs, we prompt an LLM (`gpt-5.4`) to judge whether one theorem is a meaningful extension of the other, enumerating ways that these nodes can be related through a schema drawn from prior work on mathematical development (Isaacson, 1978; Pólya, 1990; Kitcher, 1984; Thurston, 2006; Corfield, 2003) (prompt in Figure 5). The model returns an edge type which is added to the CEG. Across the three CEGs we extract 1,128 main theorems from 1,213 papers, connected by 285 main-theorem-to-main-theorem edges (and 1,255 edges in total when lemma- and proposition-level dependencies are included).

A.10 Conjecture-Generation Prompts

For each of the five context conditions described in Section 4.1, the following templates are filled at runtime with the predecessor theorem and any associated structured context, then sent verbatim to the conjecture-generation model. The five templates share the same task framing and the same output contract (a JSON list of k bare mathematical statements) so that the only difference between conditions is the structured context provided.

A.10.1 Zero-shot

A.10.2 Citation graph

A.10.3 Reduce-only (ablation)

A.10.4 Compare-only (ablation)

A.10.5 CEG (ours)

A.11 Judge prompt

Each judge model receives the prompt below, with `{predecessor_statement}`, `{conjectures_block}`, `{papers_block}`, `{k}` (number of generated conjectures), and `{n}` (number of citing follow-up papers) substituted in. The judge returns a single JSON object containing a $k \times n$ matrix of integer scores in $\{1, 2, 3\}$, which we then aggregate across the three judges by per-cell median (equivalent to majority vote, since judges are odd and scores are integer).

```
You are an expert research mathematician.
Below you will see:

  - A PREDECESSOR theorem from a
published paper.
  - {k} CANDIDATE CONJECTURES proposed
as plausible follow-up directions to the
predecessor.
  - {n} PUBLISHED PAPERS that cite the
predecessor (papers written after the
predecessor that referenced it).

Your task is to rate, for each
(conjecture, citing paper) pair, how
well the conjecture aligns with the
actual research direction taken in that
citing paper.
```

[1] Note that earlier runs used GPT-5.1; we switched to `gpt-5.4` upon its release.

```
Scoring rubric:

  1 — No meaningful relationship. The conjecture and the paper are about different things, or share only a broad area name.
  2 — Same area, but the paper addresses related-but-distinct questions or directions. There's topical overlap without alignment on what is proved/proposed.
  3 — Strong alignment. The paper substantively addresses what the conjecture proposes, or directly proves it (or a clear variant of it).

When in doubt between 2 and 3, prefer 2. Reserve 3 for cases where the paper actually does what the conjecture proposes.

PREDECESSOR THEOREM

{predecessor_statement}

CANDIDATE CONJECTURES (k = {k})

{conjectures_block}

CITING PAPERS (n = {n})

{papers_block}

TASK

Produce an alignment score for every (conjecture, citing paper) pair. Return a single JSON object of this exact shape:

{
    "scores": [
      {"c": 1, "p": 1, "s": 2},
      {"c": 1, "p": 2, "s": 1},
      ...
      {"c": {k}, "p": {n}, "s": 1}
    ]
}

Include exactly {k} × {n} entries — one per (conjecture, paper) pair.
  - "c" = conjecture index (1-indexed, matches the [C{i}] labels above)
  - "p" = paper index (1-indexed, matches the [P{j}] labels above)
  - "s" = score in {1, 2, 3}

No commentary outside the JSON.
```

A.12 Validation of LLM judges

To check whether the LLM-judge ensemble is a reasonable proxy for expert judgment, we sampled 15 randomly selected (conjecture, citing-paper) pairs spanning all four generator models and all five conditions, and asked a practicing mathematician to score each pair under the same $\{1, 2, 3\}$ rubric described in subsection A.11. One pair was skipped by the expert (the generated conjecture statement was ill-formed and could not be adjudicated), leaving 14 scored pairs.

Table 2 reports agreement between the expert and the ensemble. The ensemble achieves 92.9% within-± 1 accuracy and a mean absolute error of 0.57 on the 1–3 scale. On the headline binary question "is this a 3?", the ensemble achieves perfect recall $(4/4)$ but precision of only 0.57 $(4/7)$: every conjecture the expert judged substantively similar to a follow-up paper was also flagged by the ensemble, but the ensemble additionally flagged three pairs the expert rated 1 or 2.

Agreement metric	Value (n=14)
Exact match	50.0% (7/14)
Within ± 1	92.9% (13/14)
Mean absolute error	0.57
Precision on "3"	0.57 (4/7)
Recall on "3"	1.00 (4/4)

Table 2: Agreement between the three-judge LLM ensemble and a practicing mathematician on 14 randomly sampled (conjecture, citing-paper) pairs.

A.13 Effects of Conjecture Generation Task

For each generator × predecessor cell we record the per-predecessor fraction of follow-up papers matched (Figure 3), then pair these observations across conditions by predecessor and pool across generators. We compute a paired bootstrap 95% confidence interval on the mean paired difference (10,000 resamples, resampling (predecessor, model) units with replacement) and a Wilcoxon signed-rank test of the null hypothesis $H_0 : \mathbb{E}[\Delta] = 0$.

Section 3 reports the pooled and per-generator contrasts. The CEG vs. citation-graph comparison is significant in three of four generators individually (`gpt-5.4`, `deepseek-v3.2`, and `claude-opus-4-7`) and in the pooled test; `gemini-2.5-pro` on its own is not significant ($p = 0.29$), but its directionally positive effect contributes to the pooled result.

Generator	n	Δ **(pp)**	**95% CI**	p
`gpt-5.4`	50	$+5.61$	$[+1.76, +9.76]$	0.013
`gemini-2.5-pro`	43	$+1.29$	$[-3.19, +5.63]$	0.288
`deepseek-v3.2`	50	$+4.03$	$[-0.28, +8.59]$	0.031
`claude-opus-4-7`	50	$+6.03$	$[+1.54, +10.31]$	0.002
Pooled	193	$\mathbf{+4.35}$	$[\mathbf{+2.18}, \mathbf{+6.56}]$	$\mathbf{1.2 \times 10^{-5}}$

Table 3: Paired-bootstrap CIs and Wilcoxon signed-rank tests (two-sided) for CEG vs. citation graph, paired by predecessor and pooled across the four generator models. Three of four per-generator contrasts and the pooled test are significant.

```
Extract the main theorem from this paper along with ALL definitions, lemmas, and propositions
needed to fully understand it, structured as a dependency DAG.

Instructions:
1. Identify the paper's MAIN THEOREM (primary contribution).
2. Trace backwards to find every definition, lemma, proposition, and corollary that the main
   theorem depends on -- directly or transitively.
3. For each supporting statement, also identify THEIR dependencies within the paper.

For the MAIN THEOREM, provide its full complete statement in LaTeX.
For all SUPPORTING nodes, provide a concise one-sentence summary in LaTeX (not the full
statement -- keep it short).

Paper Title: {paper_title}

Paper Text: {paper_text}

Return as JSON:
{
    "main_theorem": {
        "id": "thm_1.1",
        "name": "Theorem 1.1",
        "statement": "Full complete statement of the main theorem in LaTeX."
    },
    "dependencies": [
        {
            "id": "def_2.1",
            "type": "definition",
            "name": "Definition 2.1",
            "summary": "One-sentence summary: a stability condition on $\mathcal{D}$ is a pair
                        $(Z, \mathcal{P})$ satisfying...",
            "depends_on": []
        },
        {
            "id": "lemma_3.4",
            "type": "lemma",
            "name": "Lemma 3.4",
            "summary": "Harder--Narasimhan filtrations exist uniquely for any stability condition.",
            "depends_on": ["def_2.1"]
        }
    ]
}

Important:
- Return ONLY valid JSON, no preamble or explanation.
- The main theorem statement must be COMPLETE -- full hypotheses and conclusions in LaTeX.
- Supporting node summaries should be SHORT (1-2 sentences). Do NOT reproduce full statements
  for supporting results.
- Use LaTeX for all math: $...$ for inline, \[...\] for display. Reconstruct proper notation
  from the raw text.
- Include ALL transitive dependencies -- if lemma A depends on definition B, include both.
- "depends_on" lists the ids of other nodes in this DAG that this node requires.
- If the paper has multiple main results, choose the most central one.
```

Figure 4: Prompt used in the reduce operation of CEG pipeline.

```
Analyze whether Theorem 2 significantly relies on or was influenced by Theorem 1.

THEOREM 1 (the earlier result, potentially depended upon):
Name: {theorem1_name}
Statement: {theorem1_statement}
Assumptions: {theorem1_assumptions}
Proof approach: {theorem1_proof}

THEOREM 2 (the later result, potentially depending on Theorem 1):
Name: {theorem2_name}
Statement: {theorem2_statement}
Assumptions: {theorem2_assumptions}
Proof approach: {theorem2_proof}

Context from Theorem 2's paper: {theorem2_context}

Determine if Theorem 2 SIGNIFICANTLY RELIES ON or was INFLUENCED BY Theorem 1.

Classify the relationship using EXACTLY ONE of the following dependency types. These categories are
derived from the mathematical philosophy literature (Lakatos 1976, Polya 1954, Gomez-Ramirez 2020,
Kitcher 1983, Thurston 1994, Corfield 2003).

DEPENDENCY TYPES:
 1. "generalization" -- Theorem 2 broadens the scope of Theorem 1 to a wider class of objects.
 2. "specialization" -- Theorem 2 restricts Theorem 1 to a particular case, obtaining sharper results.
 3. "assumption_weakening" -- Theorem 2 proves a similar result under strictly weaker hypotheses.
 4. "conclusion_strengthening" -- Theorem 2 achieves a stronger conclusion under similar hypotheses.
 5. "domain_extension" -- Theorem 2 transplants Theorem 1 into a new mathematical setting.
 6. "proof_technique_transplant" -- Theorem 2 reuses Theorem 1's proof method in a new context,
    without directly invoking Theorem 1's statement.
 7. "direct_invocation" -- Theorem 2's proof directly cites or uses Theorem 1 as a lemma or step.
 8. "analogy" -- Theorem 2 is a morally similar result in a different mathematical domain.
 9. "abstraction_unification" -- Theorem 2 abstracts away domain-specific details from Theorem 1,
    unifying them in a more general framework.
10. "counterexample_tightness" -- Theorem 2 shows Theorem 1's result is sharp / bounds are tight.
11. "reformulation" -- Theorem 2 restates Theorem 1 in a different mathematical language.
12. "lemma_incorporation" -- Theorem 2 corrects or refines Theorem 1 by making a hidden
    assumption explicit or fixing a gap.
13. "alternative_proof" -- Theorem 2 proves the same result via a fundamentally different method.
14. "none" -- No significant dependency or influence.

A significant reliance means at least one of:
- Theorem 2's proof directly uses Theorem 1 as a key step or lemma
- Theorem 2's statement builds upon, generalizes, refines, or extends Theorem 1
- Theorem 2's assumptions or conclusions modify or incorporate Theorem 1's results
- Theorem 2 reuses Theorem 1's proof strategy in a new setting
- Theorem 2 demonstrates the sharpness or limitations of Theorem 1

Do NOT consider it a dependency if:
- The theorems are merely about similar or related topics
- They share standard techniques (e.g., both use Zorn's lemma) without deeper connection
- They are in the same field but are independent results
- One is vaguely "inspired by" the other without traceable mathematical influence

Return as JSON:
{
    "has_dependency": true/false,
    "dependency_type": "<one of the 14 types above>",
    "dependency_strength": "strong" | "moderate" | "weak" | "none",
    "explanation": "Detailed explanation in LaTeX of how Theorem 2 depends on or was influenced
                    by Theorem 1. Be specific: which assumptions changed? Which proof steps
                    were reused? What was generalized and how?",
    "evidence": "Specific quotes or references from Theorem 2 in LaTeX demonstrating the dependency."
}

Be precise and mathematical. Only mark as dependency if there is clear evidence of actual
mathematical reliance or influence, not just topical similarity.
```

Figure 5: Prompt used in the compare stage to classify a candidate ordered pair (Theorem 1 = earlier, Theorem 2 = later).

```
You are an expert in mathematical research, specifically algebraic geometry and related fields.

Given the abstracts of two papers, determine whether the CITING PAPER builds upon the CITED PAPER at
the theorem level. Specifically, determine whether the citing paper contains results (theorems,
lemmas, propositions) that mathematically depend on, generalize, extend, or apply results from the
cited paper.

CITED PAPER (seed/parent):
Title: {cited_title}
Abstract: {cited_abstract}

CITING PAPER (candidate):
Title: {citing_title}
Abstract: {citing_abstract}

A theorem-level dependency means:
1. The citing paper proves a theorem that generalizes or extends a result from the cited paper.
2. The citing paper uses a theorem from the cited paper as a key step in its own proofs.
3. The citing paper's main results build upon the mathematical framework established by the
   cited paper.
4. The citing paper applies theorems from the cited paper to new settings or problems.

Do NOT consider it a theorem-level dependency if:
- The citing paper merely references the cited paper for background or motivation.
- The papers share a topic but their results are mathematically independent.
- The citing paper only uses definitions or notation from the cited paper without depending
  on its theorems.
- The relationship is purely historical or inspirational.

Return as JSON:
{
    "has_theorem_dependency": true/false,
    "confidence": "high" | "medium" | "low",
    "explanation": "Brief explanation of why this is or is not a theorem-level dependency"
}

Be precise. Only mark as having a theorem-level dependency if the abstract provides clear evidence
of mathematical reliance on the cited paper's results.
```

Figure 6: Exact prompt used for the abstract-level filter during BFS expansion. The fields {cited_title}, {cited_abstract}, {citing_title}, and {citing_abstract} are interpolated with arXiv-fetched metadata. We accept the candidate edge if the LLM returns has_theorem_dependency=true with confidence $\in$ {high, medium}; low-confidence positives are discarded.

```
You are an expert mathematician working in a specialized subfield. You will be shown a theorem
from a paper, with no additional context.

You will propose {k} conjectures which are open mathematical statements that another paper might
plausibly pursue as a next step from the predecessor theorem.

============================================================
PREDECESSOR THEOREM
============================================================

This is the predecessor theorem you are to extend:

{statement}

============================================================
TASK
============================================================

Reason only from the statement of the predecessor theorem shown above. Do not rely on any prior
context, outside knowledge, or recollection of papers, results, or follow-up work you may have
seen during training. Treat the theorem as if it is the only thing you know about its area.

Propose {k} conjectures that could plausibly be the next theorem to follow from the predecessor.
Each conjecture should be a precisely-stated open mathematical statement that another paper might
pursue.

Each conjecture should be a bare mathematical statement -- no titles, labels, numbering, or
preamble. Do not begin with "Conjecture N:" or similar.

Return a JSON list of exactly {k} strings. No commentary outside the JSON.
```

Figure 7: Zero-shot prompt template. The model sees only the predecessor statement and is explicitly instructed to avoid drawing on outside knowledge.

```
You are an expert mathematician working in a specialized subfield. You will be shown a theorem
from a paper, together with the bibliography of that paper -- each cited paper's title, year, and
abstract.

You will propose {k} conjectures which are open mathematical statements that another paper might
plausibly pursue as a next step from the predecessor theorem.

============================================================
PREDECESSOR THEOREM
============================================================

This is the predecessor theorem you are to extend:

{predecessor_statement}

============================================================
PAPERS CITED BY THE PREDECESSOR'S SOURCE PAPER
============================================================

The references below are the papers cited by the predecessor's source paper -- the intellectual
context the predecessor itself builds on.

{references_block}

============================================================
TASK
============================================================

Using everything above as context, propose {k} conjectures that could plausibly be the next
theorem to follow from the predecessor. Each conjecture should be a precisely-stated open
mathematical statement that another paper might pursue.

Each conjecture should be a bare mathematical statement -- no titles, labels, numbering, or
preamble. Do not begin with "Conjecture N:" or similar.

Return a JSON list of exactly {k} strings. No commentary outside the JSON.
```

Figure 8: Citation-graph prompt template. The `{references_block}` field is filled with up to 30 references from the Semantic Scholar `/references` endpoint, each rendered as title, year, and abstract.

```
You are an expert mathematician working in a specialized subfield. You will be shown a theorem
from a paper, together with a set of related theorems retrieved from other papers in the same
area.

You will propose {k} conjectures which are open mathematical statements that another paper might
plausibly pursue as a next step from the predecessor theorem.

============================================================
PREDECESSOR THEOREM
============================================================

This is the predecessor theorem you are to extend:

{predecessor_statement}

============================================================
RELATED THEOREMS FROM THE SURROUNDING LITERATURE
============================================================

The {n_peers} theorems below were retrieved by similarity to the predecessor above. They are
included as context for the kinds of statements that appear in this area.

{peers_block}

============================================================
TASK
============================================================

Using everything above as context, propose {k} conjectures that could plausibly be the next
theorem to follow from the predecessor. Each conjecture should be a precisely-stated open
mathematical statement that another paper might pursue.

Each conjecture should be a bare mathematical statement -- no titles, labels, numbering, or
preamble. Do not begin with "Conjecture N:" or similar.

Return a JSON list of exactly {k} strings. No commentary outside the JSON.
```

Figure 9: Reduce-only ablation prompt template. The `{peers_block}` field is filled with the top-k CEG-neighbor predecessor theorems retrieved by cosine on the predecessor statement; their successors and the edge-type labels are removed.

```
You are an expert mathematician working in a specialized subfield. You will be shown a theorem
from a paper, together with a taxonomy of the kinds of mathematical moves by which a follow-up
theorem can extend a predecessor.

You will propose {k} conjectures which are open mathematical statements that another paper might
plausibly pursue as a next step from the predecessor theorem.

============================================================
PREDECESSOR THEOREM
============================================================

This is the predecessor theorem you are to extend:

{statement}

============================================================
TAXONOMY OF MATHEMATICAL MOVES
============================================================

These are the kinds of mathematical moves a follow-up theorem might make relative to its
predecessor:

{taxonomy_block}

============================================================
TASK
============================================================

Using everything above as context, propose {k} conjectures that could plausibly be the next
theorem to follow from the predecessor. Each conjecture should be a precisely-stated open
mathematical statement that another paper might pursue.

Each conjecture should be a bare mathematical statement -- no titles, labels, numbering, or
preamble. Do not begin with "Conjecture N:" or similar. Do not label which kind of move each
conjecture corresponds to.

Return a JSON list of exactly {k} strings. No commentary outside the JSON.
```

Figure 10: Compare-only ablation prompt template. The `{taxonomy_block}` field is filled with the names and one-line descriptions of the 13 edge types in the CEG taxonomy (generalization, specialization, domain_extension, etc.); no neighboring theorem statements are shown.

```
You are an expert mathematician working in a specialized subfield. You will be shown a theorem
from a paper, together with the full local DAG of that paper (the other definitions, lemmas, and
theorems around it, and how they depend on each other), and several examples drawn from related
work in the same area. Each example shows a predecessor theorem (with the full local DAG of its
paper), the actual successor theorem that followed in the literature (with the full local DAG of
its paper), and the kind of mathematical move that linked them.

You will propose {k} conjectures which are open mathematical statements that another paper might
plausibly pursue as a next step from the predecessor theorem.

============================================================
PREDECESSOR THEOREM
============================================================

This is the predecessor theorem you are to extend:

{predecessor_statement}

----- Local DAG of the predecessor's paper -----

{predecessor_dag}

============================================================
EXAMPLES -- PREDECESSOR -> SUCCESSOR MOVES IN THIS AREA
============================================================

The {n_examples} examples below are real predecessor / successor pairs in the same area, retrieved
by similarity to the predecessor above. Each pair has been classified by the kind of mathematical
move that linked them.

{examples_block}

============================================================
TASK
============================================================

Using everything above as context, propose {k} conjectures that could plausibly be the next
theorem to follow from the predecessor. Each conjecture should be a precisely-stated open
mathematical statement that another paper might pursue.

Each conjecture should be a bare mathematical statement -- no titles, labels, numbering, or
preamble. Do not begin with "Conjecture N:" or similar. Do not label which kind of move each
conjecture corresponds to.

Return a JSON list of exactly {k} strings. No commentary outside the JSON.
```

Figure 11: CEG (full) prompt template. The `{predecessor_dag}` field is filled with the predecessor's local intra-paper DAG (definitions, lemmas, propositions, all intra-paper dependency edges). The `{examples_block}` field is filled with k_{ex} retrieved predecessor-successor pairs, each rendered as: predecessor statement and DAG, successor statement and DAG, the typed edge that linked them, and the LLM-written explanation from the corpus build.