LINEAGE: A BENCHMARK FOR EMPIRICAL LINEAGE RETRIEVAL IN AI SCIENTIFIC LITERATURE

Anonymous authors
Paper under double-blind review

ABSTRACT

Current research agents are moving beyond academic search toward literature synthesis and experiment planning. These workflows increasingly rely on literature retrieval systems, which serve as standalone systems or as components of research agents. However, existing scientific literature retrieval benchmarks mainly evaluate these methods using topical relevance or citation links as supervision, which cannot capture which prior methods are used for empirical comparison. To address this gap, we introduce **LINEAGE**, a benchmark for empirical lineage retrieval in AI scientific literature. An empirical lineage relation links a prior paper to a source paper that explicitly uses its method as an empirical comparison baseline. These relations form lineage graphs whose multi-hop neighborhoods provide retrieval ground truth for structural evaluation. To construct these graphs at scale, we develop **LINEAGE-Constructor**, which converts heterogeneous paper records and full-paper empirical baseline evidence into calibrated lineage graphs. Built from 121,048 papers across 17 top AI venues from 2018 to 2025, LINEAGE contains 3,950 graph components, 67,502 paper nodes, 164,948 lineage edges, and 80,000 multi-facet retrieval queries. Experiments with 19 representative retrieval systems show that empirical lineage retrieval remains challenging. Performance drops sharply with lineage distance, while structural metrics reveal that isolated retrieval hits often fail to recover closely connected empirical comparison neighborhoods.

1 INTRODUCTION

Large language models (LLMs) are increasingly being developed into research agents for automating scientific workflows (Lu et al., 2026; Lyu et al., 2026). These agents support different stages of research, including hypothesis formulation (Si et al., 2025; Tang et al., 2025), literature synthesis (Asai et al., 2026; Chen et al., 2026a), experiment automation (Chen et al., 2025a;d; Huang et al., 2024; Toledo et al., 2025), and manuscript generation (Song et al., 2026; Wang et al., 2024b). Across these stages, literature retrieval provides the evidence needed for downstream research. Retrieval methods can also serve as standalone systems for academic search. However, research-oriented retrieval often requires more than finding topically similar papers. For literature synthesis and experiment planning, researchers may need to identify which prior methods are used as empirical baselines and which later papers compare against a given method. These relations reveal a paper's empirical comparison context and support baseline discovery and contribution positioning. Recent studies further show that paper retrieval quality affects downstream reasoning and generation (Pan et al., 2026; Xu et al., 2026), motivating benchmarks that evaluate retrieval beyond topical relevance.

Existing scholarly retrieval benchmarks primarily build upon relevance-based or citation-oriented criteria. Relevance-based benchmarks use aspect-specific queries to retrieve papers matching a scientific statement (Ajith et al., 2024; Wang et al., 2023), but topical relevance does not imply empirical comparison. Two papers may study the same problem without being directly compared in experiments. Citation-oriented benchmarks instead use citation recommendation or prediction as supervision (He et al., 2025; Long et al., 2024; Press et al., 2024; Zheng et al., 2026), but citation relations are heterogeneous and often reflect background discussion, datasets, tools, or theoretical support. As a result, neither signal isolates which prior methods are explicitly used for empirical comparison. Existing benchmarks therefore cannot directly evaluate whether retrieval methods recover the empirical comparison context surrounding a paper.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

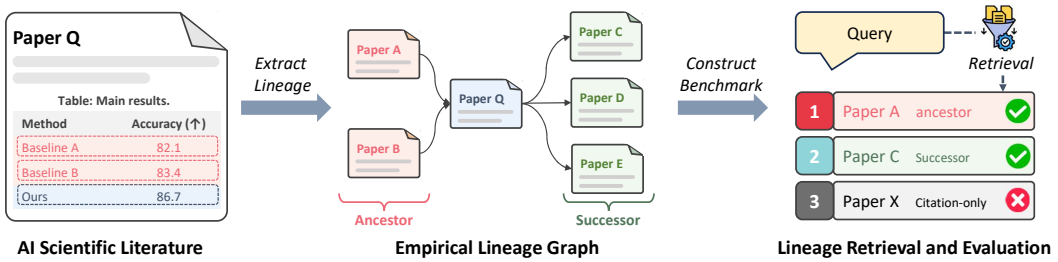


Figure 1: Illustration of empirical lineage relations. In LINEAGE, empirical baseline evidence induces directed lineage edges from baseline papers to source papers. Given a query paper, its ancestors and successors in the lineage graph define retrieval ground truth. These relations capture empirical comparison structure beyond generic topical or citation-based relevance.

To address this gap, we propose **empirical lineage** as an auditable relation for literature retrieval. As illustrated in Figure 1, an empirical lineage relation links a prior paper to a source paper that explicitly uses its method as a baseline for empirical comparison. Each relation is grounded in explicit evidence. These relations capture the empirical reference frame used to evaluate and position a contribution. By linking these relations across papers, we formulate **empirical lineage retrieval** as the task to evaluate whether a retrieval system can recover papers in these comparison-based neighborhoods.

However, constructing empirical lineage graphs from scientific literature is non-trivial. ❶ **Corpus Grounding:** Lineage retrieval requires an auditable retrieval space where each paper node and each lineage edge can be consistently identified. This is difficult because scientific papers are distributed across heterogeneous open-access sources with inconsistent coverage, metadata, and PDF availability. ❷ **Evidence-to-Edge Ambiguity:** Unlike citation links, lineage relations must be identified from full-paper empirical evidence. Baseline mentions can appear in different sections and forms, making it difficult to distinguish true comparison targets from other citations. ❸ **Graph Suitability:** Extracted lineage relations do not directly form graphs suitable for retrieval evaluation. Some widely reused baselines can create overly broad connected regions even when individual comparison relations are limited. Such regions make multi-hop neighborhoods less suitable as retrieval ground truth.

To address these challenges, we propose **LINEAGE-Constructor**, an automatic pipeline for constructing empirical lineage graphs from scientific literature. The pipeline consists of three stages. ❶ **Unified Literature Collector** builds a unified retrieval corpus from heterogeneous open-access sources, aligning metadata, validating PDFs, and normalizing paper records. ❷ **Evidence-Grounded Lineage Extraction** extracts lineage relations from empirical baseline evidence. Since such evidence can appear across different parts in varying formats, we design a dual-source extractor that provides complementary paper representations and enables cross-checking of outputs, improving coverage of dispersed baseline mentions while reducing spurious edges. We further annotate a human-annotated validation set, LINEAGE-HA, to guide extractor optimization and evaluate extraction quality. ❸ **Graph Calibration** enforces temporal validity and regularizes abnormal graph connectivity, yielding calibrated lineage graphs for retrieval evaluation.

Based on these calibrated graphs, we build the **LINEAGE** benchmark for empirical lineage retrieval in artificial intelligence (AI) scientific literature. As summarized in Table 1, LINEAGE is, to the best of our knowledge, the first large-scale literature retrieval benchmark using empirical lineage relations as retrieval ground truth. For each query paper, LINEAGE generates *multi-facet queries* that describe the same paper from different retrieval intents, using its ancestor and successor papers in the lineage graph as retrieval ground truth. Beyond standard ranking metrics such as recall and nDCG, LINEAGE introduces structural metrics that measure whether retrieved papers recover coherent empirical comparison neighborhoods rather than isolated lineage hits.

The LINEAGE benchmark comprises **121,048 papers** from **17 top AI conference venues** published between 2018 and 2025. From this corpus, LINEAGE-Constructor constructs **3,950 directed lineage graph components**, encompassing 67,502 papers and 164,948 lineage edges. Based on these graphs, we generate **80,000 multi-facet retrieval queries**. Experimental results highlight the substantial challenges in empirical lineage retrieval: (1) Even the strongest agentic retriever achieves only 31.87 Recall@20 and 9.92 Precision@20, indicating that current retrievers fail to comprehensively recover empirical lineage targets. (2) Performance degrades rapidly with lineage distance, with average

Table 1: Comparison with relevance-based and citation-oriented literature retrieval benchmarks.

Benchmark	Ground Truth	Multi-Facet Query	Automatic Construction	Lineage Relation	Multi-Hop Retrieval	Structural Metrics
LitSearch	Relevant Paper	✗	✗	✗	✗	✗
CiteME	Citation	✓	✗	✗	✗	✗
DORIS-MAE	Relevant Paper	✓	✗	✗	✗	✗
CitationR	Citation	✗	✓	✗	✗	✗
PaSa	Citation	✗	✓	✗	✗	✗
CiteRAG	Citation	✓	✓	✗	✗	✗
LINEAGE (Ours)	Lineage Relation	✓	✓	✓	✓	✓

Recall@20 decreasing from 28.42 for one-hop targets to 6.68 for three-hop targets. (3) Graph-based metrics further reveal that occasional retrieval hits do not imply recovery of lineage structure, as online search engines achieve non-trivial hit rates but much lower structural scores.

Our contributions are three-fold: 1) We propose **empirical lineage relations** as a ground-truth signal for benchmarking literature retrieval, enabling the evaluation of whether retrieval methods recover experiment-grounded empirical neighborhoods beyond topical relevance and citation connectivity. 2) We design **LINEAGE-Constructor**, an automatic pipeline that builds lineage graphs by extracting empirical baseline evidence from full papers. The pipeline integrates a unified literature collector, evidence-grounded lineage extraction, and temporal and structural lineage graph calibration to improve the quality of the constructed lineage graphs. 3) We introduce **LINEAGE**, a large-scale benchmark featuring multi-facet queries and structural metrics, to systematically evaluate representative retrieval systems, revealing substantial gaps in capturing empirical lineage and highlighting opportunities for future research.

2 Related Work

Academic research benchmarks evaluate AI systems across scholarly workflows, including document processing, scientific document understanding, literature retrieval, citation-oriented recommendation, scientific question answering, and broader research-agent evaluation (Chen et al., 2025a; Cohan et al., 2020; Dasigi et al., 2021; Huang et al., 2024; Lo et al., 2020; Singh et al., 2023; Wadden et al., 2020). Recent work further narrows this focus to literature retrieval, examining fine-grained scenarios such as aspect-based scientific queries, complex literature search, citation-grounded retrieval, and retrieval over textual or relational scientific knowledge bases (Ajith et al., 2024; Press et al., 2024; Shao et al., 2025a; Wang et al., 2023; Wu et al., 2024; Zheng et al., 2026). Meanwhile, LLM-based research agents increasingly rely on literature retrieval to support research ideation, experiment planning, survey generation, manuscript writing, and academic search (He et al., 2025; Pan et al., 2026; Si et al., 2025; Song et al., 2026; Toledo et al., 2025; Wang et al., 2024b; Weng et al., 2025). However, their evaluations typically emphasize topical relevance, citation-based association, evidence support, or downstream research-oriented generation. In contrast, LINEAGE targets **empirical lineage retrieval**. It uses explicit empirical baseline relations as retrieval ground truth.

3 LINEAGE-Constructor

The LINEAGE-Constructor pipeline involves three major modules: unified scientific literature collector, evidence-grounded lineage extraction, and lineage graph calibration, as illustrated in Figure 2. These modules enable efficient automated construction with traceable quality, thereby significantly reducing construction costs, as discussed in §F.3.

3.1 Unified Scientific Literature Collector

Constructing empirical lineage graphs requires literature records that are both comprehensive and consistently represented. Unlike citation-based retrieval benchmarks, which rely on a single paper, lineage retrieval needs broad coverage across AI venues, as each directed edge points from the

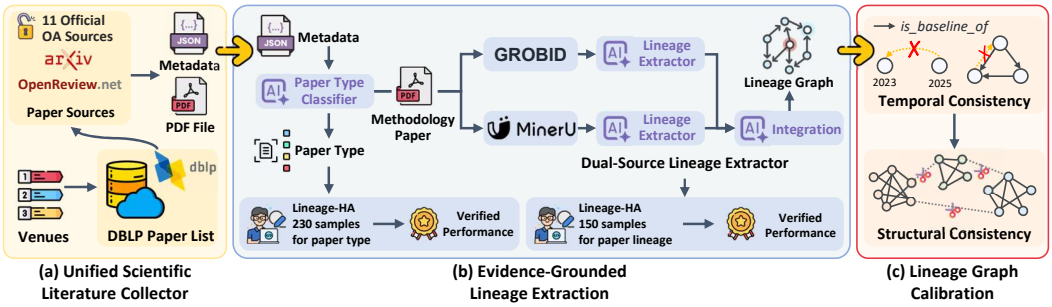


Figure 2: Overview of the LINEAGE-Constructor pipeline, consisting of (a) unified scientific literature collector, (b) evidence-grounded lineage extraction, and (c) lineage graph calibration.

baseline paper to the source paper that uses it as an empirical comparison baseline. Literature collection is, therefore, a core component of benchmark construction.

In practice, there is no single open-access source providing complete and uniform records. DBLP (DBLP) offers reliable venue-year paper lists, but abstracts and PDFs often need to be fetched from other providers. These providers differ in metadata schema, naming conventions, URL formats, and file availability. Directly merging them may create inconsistent records and noisy classifiers, which can propagate errors into lineage extraction and graph construction.

We built a unified literature collector module to address this issue. The module integrates 11 complementary open-access sources. It first constructs a canonical venue-year paper list from DBLP and filters out records that are unlikely to be regular research papers, including tutorials, demos, workshops, and student abstracts. It then enriches each retained paper using venue-specific open-access sources. Official proceedings are prioritized when available. OpenReview (OpenReview) and arXiv (arxiv) are used to recover missing PDFs. The venue-specific source configuration is provided in §F.1. We restrict the benchmark corpus to peer-reviewed conference papers, as they commonly provide explicit empirical comparisons and consistent full-paper evidence for lineage extraction.

After metadata enrichment, the module downloads and validates the corresponding PDFs before downstream parsing. A lightweight page-length filter removes short files that are unlikely to be regular papers. The module also performs diagnostic checks for suspicious metadata and PDFs, as detailed in §F.2. These checks provide a practical repair interface and reduce the risk of noisy records propagating into the lineage construction process.

3.2 EVIDENCE-GROUNDED LINEAGE EXTRACTION

Unlike citation links, which are explicitly marked and often available from structured metadata, empirical lineage relations must be identified from full-paper comparison evidence. These signals are distributed across papers and often use non-canonical method names, making direct rule-based construction unreliable. We therefore design an evidence-grounded lineage extraction module that progressively narrows the corpus, grounds extracted relations in paper content, and converts the retained relations into a corpus-grounded lineage graph.

Paper Type Classifier. Regular papers in our corpora include various contribution types, such as methodology papers, benchmark papers, and surveys. Since benchmark and survey papers primarily build evaluation resources or synthesize existing work rather than proposing new technical methods, their citation relations are less suitable for constructing lineage graphs. We therefore first identify methodology papers for downstream lineage extraction. Specifically, we prompt `Qwen3.5-27B`[1] with each paper's title and abstract. The prompt classifies papers by their primary contribution, classifying papers as *methodology* only when the central contribution is a technical method rather than an evaluation resource or a synthesis of prior work. Detailed prompts are provided in §I.1.

Dual-Source Lineage Extractor. To robustly extract lineage relations from scientific papers, we propose a dual-source lineage extractor with candidate-restricted integration. Each PDF is parsed by two complementary tools: `GROBID` (gro, 2008), which produces semi-structured XML, and

[1] https://huggingface.co/Qwen/Qwen3.5-27B

MinerU (Wang et al., 2024a), which produces markdown with paper-related structures. Rather than concatenating the two parser outputs, we prompt Qwen3.5-27B to **extract** candidate lineage relations from each source independently with the prompt presented in §I.1. We then **integrate** the two candidate sets through a grounding step that re-checks each candidate against the paper content and retains only supported relations. This design uses parser-specific extraction to improve recall while limiting integration to evidence-based conflict resolution. The model also normalizes cited paper titles and removes duplicates.

LINEAGE-HA. To evaluate the reliability of our automated extraction module, we construct LINEAGE-HA, a human-annotated validation set containing 230 papers annotated for paper type classification and 150 papers annotated for lineage relation extraction. The lineage-relation annotations contain 970 positive empirical lineage relations. It allows us to evaluate whether the module correctly filters methodology papers and extracts supported lineage relations from full papers. Details of LINEAGE-HA are provided in §E, where the results show that our paper type classification and dual-source lineage extraction outperform their alternatives.

Lineage Graph Construction. After extracting lineage relations from individual papers, we construct a corpus-grounded directed lineage graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ over the collected paper database. Each node $v \in \mathcal{V}$ denotes a paper in the collected corpus, and each directed edge $e = (v_b, v_s) \in \mathcal{E}$ indicates that the source paper v_s explicitly uses the method from v_b as an empirical comparison baseline. We retain only source–baseline pairs in which both papers can be matched to the collected corpus, so that every graph node belongs to the same retrieval space used for downstream evaluation. Nodes store basic paper metadata, while edges retain extraction provenance. The resulting graph encodes "is_baseline_of" relations, with the reverse direction corresponding to "uses_baseline".

3.3 LINEAGE GRAPH CALIBRATION

To enhance the reliability of constructed lineage graphs, we employ temporal and structural consistency calibration to filter out abnormal relations. Temporal calibration enforces valid edge ordering. Structural calibration controls overly broad connectivity in multi-hop neighborhoods. Stage-wise calibration analysis is in § F.2.

Temporal Consistency. We first calibrate the graph by enforcing temporal validity. A retained lineage edge must point from a baseline paper to a source paper published in the same year or later, since an empirical baseline cannot depend on future work. Edges that violate this ordering are removed as invalid lineage candidates.

Structural Consistency. We further calibrate the graph to make its connected regions suitable for lineage-based retrieval evaluation. Because widely reused baselines can create very large connected components, raw components can be too broad for defining informative multi-hop retrieval neighborhoods. We therefore apply a fixed component-level calibration procedure to split components classified as upper-tail outliers by the IQR rule (Chatfield, 1986) into more coherent subgraphs using bridge pruning (Tarjan, 1974) and Louvain community partitioning (Blondel et al., 2008). Finally, we discard isolated nodes that no longer participate in any retained lineage relation. These operations produce lineage graphs with valid temporal ordering and controlled connectivity for retrieval evaluation.

4 DATASET

Based on the calibrated lineage graphs constructed by LINEAGE-Constructor, we introduce LINEAGE, a benchmark for empirical lineage retrieval. It covers papers from 17 top AI venues from 2018 to 2025, including 121,048 papers with complete metadata records, as detailed in §G.1. After lineage extraction and calibration, we construct 80,000 multi-facet retrieval queries as the **LINEAGE benchmark**. It supports evaluation with both standard retrieval metrics and graph-structure metrics.

Specifically, given a calibrated directed lineage graph $G = (V, E)$, LINEAGE evaluates each query paper $v_q \in V$ by generating multi-facet retrieval queries. A retrieval method $\mathcal{M}$ returns a ranked list $R_k(q, \mathcal{M})$ from our corpus $\mathcal{C}$ (*i.e.,* the collected 121,048 papers), and the retrieved papers are compared with the empirical lineage ground-truth papers Y_q, which are derived from the ancestors and successors of v_q.

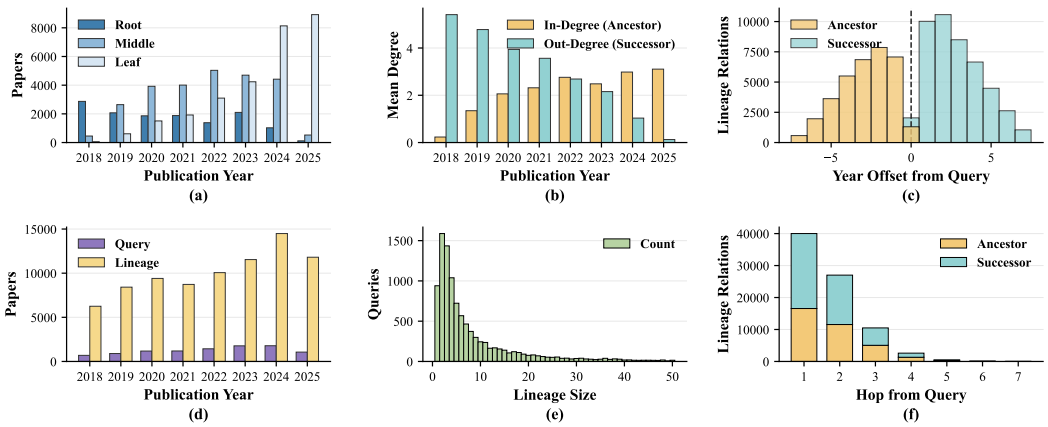


Figure 3: Graph statistics (a-c) and query properties (d-f) in LINEAGE. (a) Publication-year distribution of root, middle, and leaf nodes. (b) Average in-degree and out-degree by publication year. (c) Year offsets between query papers and their ancestor or successor ground-truth papers. (d) Yearly distribution of query papers and lineage papers. (e) Distribution of ground-truth lineage set size per query. (f) Distribution of ground-truth lineage papers by hop distance from the query paper.

4.1 LINEAGE GRAPH

In LINEAGE, each calibrated lineage graph contains papers connected through empirical baseline relations. Nodes correspond to papers, and directed edges indicate explicit empirical comparison relations. We orient each edge from a baseline paper to the source paper that uses its method for empirical comparison. Under this direction, ancestors trace earlier papers connected through one or more baseline relations, while successors trace later papers connected in the opposite direction. As shown in Table 2, LINEAGE contains 3,950 lineage graphs, 67,502 paper nodes, and 164,948 lineage edges. The graph collection includes 13,315 root nodes and 28,488 leaf nodes, covering both starting-point and terminal papers under the current observation window.

Table 2: LINEAGE and query statistics.

Domain	Statistic	Value
Lineage Graph	Component	3,950
	#Node (*Paper*)	67,502
	#Root	13,315
	#Leaf	28,488
	#Middle	25,699
	#Edge (*Lineage*)	164,948
Multi-Facet Queries	#Query papers	10,000
	#Query	80,000
	#Root	26,672
	#Leaf	26,664
	#Middle	26,664
	Lineage/Query	8.1

Figures 3(a)–(c) illustrate the temporal structure of the constructed lineage graph. At the graph level, root nodes concentrate in earlier years, whereas leaf nodes become more common in recent years, as shown in Figure 3(a). This node-level tendency is complemented by the degree statistics in Figure 3(b), where earlier papers have larger successor out-degree and recent papers have larger ancestor in-degree. The relation timing gives a more direct view of this ordering. Ancestors lie mainly before the paper under consideration, and successors mainly after it, as reported in Figure 3(c). These distributions show that the constructed empirical lineage graphs follow the expected temporal ordering. Additional statistics and cases are provided in §G.2 and §H.1.

4.2 MULTI-FACET QUERIES

Common literature search behaviors are heterogeneous across users and retrieval methods. To cover these scenarios, we generate eight multi-facet retrieval queries for each query paper by prompting an LLM to extract and reformulate salient information from the title and abstract by the prompt illustrated in §I.2. Specifically, the ***title*** query uses the paper title directly and targets exact paper lookup, while the ***title-context*** query concatenates the title and abstract to provide more detailed information. The ***topical keyword*** query extracts a short set of keywords, while the ***topical semantic*** query rephrases the problem as a concise query. We further generate keywords and semantic variants for two additional facets. The contextual facet focuses on the background, yielding ***context keyword*** and ***context semantic*** queries. The methodological facet focuses on the technical mechanism, yielding ***method keyword*** and ***method semantic*** queries. §H.2 illustrates the cases for each kind of query.

Table 3: Evaluation results of representative retrieval methods on LINEAGE. The best result in each metric is highlighted in **bold**, and the second-best result is underlined.

Category	Retrieval Method	Retrieval Metric					Structural Metric		
		R@20	*P*@20	*H*@20	*N*@20	*M*@20	*HWR*@20	*DR*@20	*GN*@20
w/ LINEAGE *Corpus* (Avg.)		24.82	7.34	54.95	19.54	27.34	26.01	28.42	19.70
Sparse Retrieval	BM25	22.06	6.48	52.81	17.56	25.90	23.28	25.74	17.88
	TF-IDF	20.42	6.02	49.90	15.81	22.95	21.48	23.64	16.00
Dense Retrieval	Contriever	18.20	5.24	46.91	14.29	21.52	19.18	21.19	14.48
	BGE-M3 (0.6B)	10.71	3.03	32.54	8.30	13.43	11.24	12.34	8.37
	Jina-Embedding-v5 (0.6B)	26.26	7.58	57.14	20.50	28.48	27.64	30.46	20.83
	Qwen3-Embedding (0.6B)	25.96	7.69	56.74	20.40	28.31	27.34	30.18	20.71
	Qwen3-Embedding (4B)	30.64	8.92	61.88	23.90	31.92	32.10	35.09	24.10
	Qwen3-Embedding (8B)	29.35	8.52	60.32	22.78	30.54	30.60	33.18	22.80
	NV-Embedding-v2 (8B)	29.76	8.87	61.21	23.14	30.92	31.28	34.38	23.41
	ReasonIR (8B)	25.38	7.35	57.03	19.72	27.71	26.60	29.11	19.88
Agentic Retrieval	Pasa	20.41	6.07	49.50	17.06	25.74	21.41	23.47	17.27
	AgentIR$^\diamond$	27.61	8.29	60.12	22.15	31.31	28.85	31.43	22.28
	AgentIR-Qwen3-Embedding$^\diamond$	28.78	8.82	59.25	22.93	30.98	29.88	31.81	22.82
	Tongyi-DeepResearch$^\diamond$	**31.87**	**9.92**	**63.88**	**24.99**	**33.09**	**33.23**	**35.89**	**24.99**
w/ Online Sources (Avg.)		6.69	1.90	24.95	5.62	10.31	6.89	5.59	4.33
Search Engine	Google Search$^\diamond$	9.64	0.40	32.37	6.64	10.62	9.99	2.23	1.42
	Google Scholar$^\diamond$	7.57	2.00	23.62	5.23	8.11	8.01	8.78	5.14
LLM Search	GPT-5.4$^\diamond$	7.24	3.30	29.13	7.12	13.56	7.42	7.87	6.65
	Gemini-3.1$^\diamond$	4.47	1.97	21.50	4.58	10.35	4.51	4.46	4.23
	DeepSeek-V3.2$^\diamond$	4.52	1.82	18.12	4.55	8.92	4.51	4.59	4.23

$^\diamond$ Due to the rate limit of the API, we evaluate these methods on a sampled subset. The details of the subset and other methods' performance on it are presented in §D.1.

To reduce noisy queries, we use `Jina-Embedding-v5-0.6B` (Akram et al., 2026) to retrieve the query paper from which each query is generated. If the query paper is not ranked among the top 20 results, the query is regenerated. As shown in Table 2, the selected query papers are balanced across root, leaf, and middle nodes. The query set maintains broad temporal coverage, with its yearly distribution closely following that of lineage papers in Figure 3(d). The retrieval ground-truth sets are also heterogeneous, as their size follows a long-tailed distribution in Figure 3(e). Beyond variable target sizes, the hop distribution spans multiple graph distances, as shown in Figure 3(f). It covers both direct empirical comparison neighbors and papers connected through longer lineage paths, enabling evaluation across a broader structural range. We further report the statistics of queries and ground-truth in §G.3.

4.3 EVALUATION METRICS

In LINEAGE, we evaluate retrieval using both standard and graph-structural metrics.

Retrieval. Standard retrieval metrics, including **R**ecall@k, **P**recision@k, **H**it Rate@k, **N**DCG@k, and **M**RR@k, measure whether lineage ground-truth papers are retrieved and whether they appear in high-ranking positions. Since the ground truth is an unordered set of lineage papers, **N**DCG@k is computed with binary relevance.

Structural. Since standard metrics do not capture retrieval performance over the lineage graph structure, we introduce several structural metrics for lineage retrieval. Graph distance reflects the length of the empirical comparison chain connecting a target paper to the query paper. **H**op-**W**eighted **R**ecall@k assigns larger weights to papers connected through shorter paths. **D**irect **R**ecall@k focuses on one-hop lineage neighbors, which correspond to explicit empirical baseline relations. **G**raph-weighted **n**DCG@k further evaluates whether structurally closer lineage papers are ranked higher.

5 RESULTS

In experiments, we evaluate 19 retrieval methods on LINEAGE under two settings. The corpus-based setting includes 14 methods that retrieve from the fixed LINEAGE corpus, covering two sparse retrievers based on lexical matching, eight dense retrievers based on embedding similarity, and four agentic retrieval methods based on iterative search and candidate selection. The online-source setting

includes five methods, covering two search engines and three LLMs enforced by online search. Detailed model choices and implementation settings are provided in §C.2.

The corpus-based baselines span three retrieval paradigms. Sparse retrievers include BM25 (Robertson & Zaragoza, 2009) and TF-IDF (Sparck Jones, 1972), which rely on lexical matching. Dense retrievers include Contriever (Izacard et al., 2022), BGE-M3 (Chen et al., 2024), Jina-Embedding-v5 (Akram et al., 2026), Qwen3-Embedding (0.6B, 4B, and 8B) (Zhang et al., 2025), NV-Embedding-v2 (Lee et al., 2025), and ReasonIR (Shao et al., 2025b), which rank papers by embedding similarity. Agentic retrievers include PaSa (He et al., 2025), AgentIR and AgentIR-Qwen3-Embedding (Chen et al., 2026b), and Tongyi-DeepResearch (Team et al., 2025), which iteratively reformulate queries and select candidates.

5.1 Main Results

Table 3 reveals several findings about the difficulty of retrieving lineage-related papers and recovering empirical lineage structure in AI scientific literature.

Lineage retrieval remains challenging. The main results show that empirical lineage retrieval remains far from solved. Even the strongest method, Tongyi-DeepResearch, achieves only 31.87 R@20 and 9.92 P@20. Across categories, sparse retrievers provide competitive lexical baselines, dense retrievers improve performance with stronger semantic representations, and agentic retrievers achieve the best overall results. However, none of these categories retrieves lineage papers comprehensively. It indicates that current retrieval methods can sometimes identify relevant lineage papers, but still struggle to capture the full set of papers connected through empirical comparison associated with a query paper.

Agentic retrieval improves performance but remains insufficient. Although agentic retrieval achieves the best overall performance, Pasa performs worse than BM25 on the full benchmark. These results suggest that current academic search agents primarily improve query reformulation and candidate selection, but still lack explicit mechanisms for tracing empirical comparison relations.

Structural metrics reveal failures hidden by standard retrieval metrics. The structural metrics expose retrieval failures that are not fully captured by standard ranking metrics. For example, Google Search obtains a non-trivial H@20, but its DR@20 and GN@20 remain very low. This shows that occasional hits do not imply recovery of the local empirical comparison structure.

A well-crafted paper source is necessary for lineage retrieval. The gap between corpus-based and online-source retrieval is substantial, with average R@20 dropping from 24.82 to 6.69 and average DR@20 dropping from 28.42 to 5.59. Although online-source methods are restricted to academic sources and the same publication years, they still struggle to consistently retrieve lineage papers. This suggests that lineage retrieval relies on a well-crafted paper source, highlighting the need for a clean, comprehensive, and stable data interface for evaluating empirical lineage retrieval.

Ancestor retrieval is easier than successor retrieval. Figure 4 shows that most methods achieve higher R@20 on ancestors than successors. Dense retrievers exhibit noticeable variation in directional preference. The gap is especially large for agentic and LLM-based methods, suggesting that they struggle to identify later papers that use a given method as an empirical baseline.

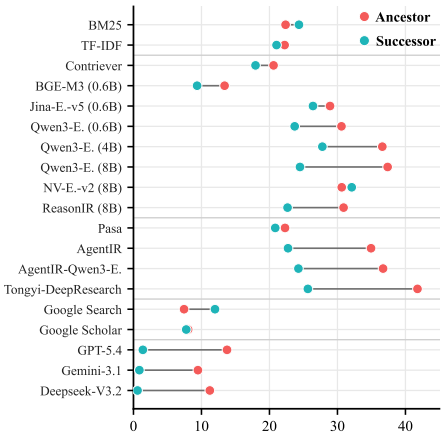


Figure 4: Recall@20 for ancestor and successor retrieval across methods.

5.2 Retrieval Performance across Query Types

To investigate how query representations affect lineage retrieval, we report category-level *R*@20 and *HWR*@20 across the eight query types in Table 4. Method-level results are provided in §D.3. Retrieval performance varies substantially across query types. *Title-Ctx* performs best for sparse, dense, and

Table 4: Performance of retrieval categories across different query types. For each category, the best query type is highlighted in **bold**, and the second-best query type is underlined.

Category	Metric	*Title*	*Title-Ctx*	*Top-KW*	*Top-Sem*	*Ctx-KW*	*Ctx-Sem*	*Meth-KW*	*Meth-Sem*
Sparse Retrieval	*R*@20	25.60	**36.58**	24.66	25.41	15.52	20.75	7.03	14.37
	HWR@20	26.96	**38.79**	25.95	26.82	16.19	21.71	7.42	15.18
Dense Retrieval	*R*@20	29.44	**39.71**	28.19	30.25	18.18	25.25	7.42	17.80
	HWR@20	30.96	**41.97**	29.61	31.84	18.88	26.28	7.78	18.69
Agentic Retrieval	*R*@20	34.57	**38.92**	33.46	35.53	22.00	30.39	7.73	17.19
	HWR@20	36.20	**40.84**	35.00	37.27	22.65	31.31	8.04	18.05
Search Engine	*R*@20	8.27	1.02	**16.31**	13.41	8.45	9.97	4.00	7.38
	HWR@20	8.51	1.14	**17.36**	14.16	8.64	10.04	4.22	7.92
LLM Search	*R*@20	6.73	**7.46**	6.15	4.98	5.51	5.05	3.05	4.33
	HWR@20	6.90	**7.62**	6.13	5.16	5.61	5.02	3.01	4.41

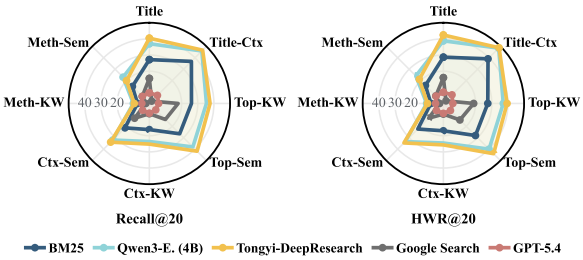


Figure 5: Methods' performance over query types.

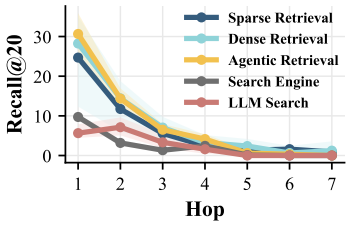


Figure 6: Recall@20 across lineage hops.

agentic retrievers, and also achieves the best result for LLM search. This suggests that combining the paper title with abstract-level context provides both paper-specific anchors and sufficient semantic information for retrieving lineage-related papers. Topical queries are also effective. *Top-Sem* ranks second for dense and agentic retrievers, while *Top-KW* performs best for search engines. In contrast, context-only and method-only queries are generally weaker, with *Meth-KW* performing worst across categories. Furthermore, Figure 5 shows distinct query sensitivity across retrieval paradigms. BM25, Qwen3-E. (4B), and Tongyi-DeepResearch perform best on *Title-Ctx* and *Top-Sem*, while Google Search favors *Top-KW*, suggesting that web search benefits more from concise topical keywords than from longer paper-level descriptions. GPT-5.4 remains weak and relatively insensitive to query formulation, indicating that query reformulation alone does not substantially improve its lineage retrieval capability. These results show that query formulation remains an important bottleneck even for stronger retrieval methods.

5.3 Retrieval Performance across Lineage Hops

To examine whether retrieval methods can trace lineage papers beyond direct empirical comparison relations, we further group ground-truth papers by their shortest-path distance from the query paper and report category-level R@20 in Figure 6. Method-level results are provided in §D.2. It shows that lineage distance substantially increases retrieval difficulty. The retrieval performance decreases rapidly as hop distance increases. These results highlight a key limitation of current agentic retrieval methods. While query reformulation and candidate selection improve their local retrieval performance, they do not explicitly traverse empirical lineage across papers. These results suggest that future agentic retrieval methods may benefit from explicit lineage expansion or lineage-aware supervision to better trace long-range empirical comparison chains.

6 Conclusion

In this paper, we introduced Lineage, a benchmark for evaluating empirical lineage retrieval in AI scientific literature. Lineage uses empirical lineage relations as retrieval ground truth and evaluates whether retrieval methods can identify empirically connected papers. Our experiments show that existing retrieval methods still struggle to comprehensively retrieve lineage papers, highlighting the challenge of empirical lineage retrieval for future literature retrieval systems.

GENERATIVE AI USE DISCLOSURE

We used generative AI for automated data annotation, including paper type classification and evidence-grounded lineage relation extraction (§3.2), and for retrieval query generation (§4.2). Annotation quality was evaluated against the human-annotated LINEAGE-HA validation set (Appendix E). We also used generative AI to check the manuscript for potential grammatical and formatting errors. The authors take responsibility for the final content of this work, including all AI-assisted text and artifacts.

REFERENCES

Grobid. https://github.com/grobidOrg/grobid, 2008.

Anirudh Ajith, Mengzhou Xia, Alexis Chevalier, Tanya Goyal, Danqi Chen, and Tianyu Gao. Litsearch: A retrieval benchmark for scientific literature search. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 15068–15083, 2024.

Mohammad Kalim Akram, Saba Sturua, Nastia Havriushenko, Quentin Herreros, Michael Günther, Maximilian Werk, and Han Xiao. jina-embeddings-v5-text: Task-targeted embedding distillation. *arXiv preprint arXiv:2602.15547*, 2026.

arxiv. arXiv. https://arxiv.org, 2026.

Akari Asai, Jacqueline He, Rulin Shao, Weijia Shi, Amanpreet Singh, Joseph Chee Chang, Kyle Lo, Luca Soldaini, Sergey Feldman, Mike D'Arcy, et al. Synthesizing scientific literature with retrieval-augmented language models. *Nature*, pp. 1–7, 2026.

Joydeep Biswas, Sheila Schoepp, Gautham Vasan, Anthony Opipari, Arthur Zhang, Zichao Hu, Sebastian Joseph, Matthew Lease, Junyi Jessy Li, Peter Stone, et al. Ai-assisted peer review at scale: The aaai-26 ai review pilot. *arXiv preprint arXiv:2604.13940*, 2026.

Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008 (10):P10008, 2008.

Chris Chatfield. Exploratory data analysis. *European journal of operational research*, 23(1):5–13, 1986.

Hui Chen, Miao Xiong, Yujie Lu, Wei Han, Ailin Deng, Yufei He, Jiaying Wu, Yibo Li, Yue Liu, and Bryan Hooi. Mlr-bench: Evaluating ai agents on open-ended machine learning research. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2025a.

Jianlyu Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. M3-embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. In *Findings of the association for computational linguistics: ACL 2024*, pp. 2318–2335, 2024.

Jiefeng Chen, Bhavana Dalvi Mishra, Jaehyun Nam, Rui Meng, Tomas Pfister, and Jinsung Yoon. Mars: Modular agent with reflective search for automated ai research. *arXiv preprint arXiv:2602.02660*, 2026a.

Qiguang Chen, Zheng Yan, Mingda Yang, Libo Qin, Yixin Yuan, Hanjing Li, Jinhao Liu, Yiyan Ji, Dengyun Peng, Jiannan Guan, et al. Autopr: Let's automate your academic promotion! *arXiv preprint arXiv:2510.09558*, 2025b.

Zijian Chen, Xueguang Ma, Shengyao Zhuang, Ping Nie, Kai Zou, Sahel Sharifymoghaddam, Andrew Liu, Joshua Green, Kshama Patel, Ruoxi Meng, et al. Browsecomp-plus: A more fair and transparent evaluation benchmark of deep-research agent. In *First Workshop on Multi-Turn Interactions in Large Language Models*, 2025c.

Zijian Chen, Xueguang Ma, Shengyao Zhuang, Jimmy Lin, Akari Asai, and Victor Zhong. Agentir: Reasoning-aware retrieval for deep research agents. *arXiv preprint arXiv:2603.04384*, 2026b.

Ziru Chen, Shijie Chen, Yuting Ning, Qianheng Zhang, Boshi Wang, Botao Yu, Yifei Li, Zeyi Liao, Chen Wei, Zitong Lu, et al. Scienceagentbench: Toward rigorous assessment of language agents for data-driven scientific discovery. In *The Thirteenth International Conference on Learning Representations*, 2025d.

Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel S Weld. Specter: Document-level representation learning using citation-informed transformers. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pp. 2270–2282, 2020.

Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A Smith, and Matt Gardner. A dataset of information-seeking questions and answers anchored in research papers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4599–4610, 2021.

DBLP. DBLP Computer Science Bibliography. `https://dblp.org`, 2026.

Yichen He, Guanhua Huang, Peiyuan Feng, Yuan Lin, Yuchen Zhang, Hang Li, et al. Pasa: An llm agent for comprehensive academic paper search. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11663–11679, 2025.

Qian Huang, Jian Vora, Percy Liang, and Jure Leskovec. Mlagentbench: Evaluating language agents on machine learning experimentation. In *International Conference on Machine Learning*, pp. 20271–20309. PMLR, 2024.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. Unsupervised dense information retrieval with contrastive learning. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856.

Zhuohang Jiang, Pangjing Wu, Ziran Liang, Peter Q Chen, Xu Yuan, Ye Jia, Jiancheng Tu, Chen Li, Peter HF Ng, and Qing Li. Hibench: Benchmarking llms capability on hierarchical structure reasoning. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 5505–5515, 2025.

Dongyeop Kang, Waleed Ammar, Bhavana Dalvi, Madeleine Van Zuylen, Sebastian Kohlmeier, Eduard Hovy, and Roy Schwartz. A dataset of peer reviews (peerread): Collection, insights and nlp applications. In *Proceedings of the 2018 conference of the North American chapter of the Association for Computational Linguistics: Human language technologies, volume 1 (long papers)*, pp. 1647–1661, 2018.

Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. NV-embed: Improved techniques for training LLMs as generalist embedding models. In *The Thirteenth International Conference on Learning Representations*, 2025.

Yoonjoo Lee, Kyungjae Lee, Sunghyun Park, Dasol Hwang, Jaehyeon Kim, Hong-in Lee, and Moontae Lee. Qasa: advanced question answering on scientific articles. In *International Conference on Machine Learning*, pp. 19036–19052. PMLR, 2023.

Chuhan Li, Ziyao Shangguan, Yilun Zhao, Deyuan Li, Yixin Liu, and Arman Cohan. M3sciqa: A multi-modal multi-document scientific qa benchmark for evaluating foundation models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 15419–15446, 2024.

Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Daniel S Weld. S2orc: The semantic scholar open research corpus. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pp. 4969–4983, 2020.

Kehan Long, Shasha Li, Pancheng Wang, Chenlong Bao, Jintao Tang, and Ting Wang. Recommending missed citations identified by reviewers: A new task, dataset and baselines. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pp. 13699–13711, 2024.

Chris Lu, Cong Lu, Robert Tjarko Lange, Yutaro Yamada, Shengran Hu, Jakob Foerster, David Ha, and Jeff Clune. Towards end-to-end automation of ai research. *Nature*, 651(8107):914–919, 2026.

Yougang Lyu, Xi Zhang, Xinhao Yi, Yuyue Zhao, Shuyu Guo, Wenxiang Hu, Jan Piotrowski, Jakub Kaliski, Jacopo Urbani, Zaiqiao Meng, et al. Evoscientist: Towards multi-agent evolving ai scientists for end-to-end scientific discovery. *arXiv preprint arXiv:2603.08127*, 2026.

Bodhisattwa Prasad Majumder, Harshit Surana, Dhruv Agarwal, Bhavana Dalvi Mishra, Abhijeetsingh Meena, Aryan Prakhar, Tirth Vora, Tushar Khot, Ashish Sabharwal, and Peter Clark. Discoverybench: Towards data-driven discovery with large language models. *arXiv preprint arXiv:2407.01725*, 2024.

J-P Onnela, Jari Saramäki, Jorkki Hyvönen, György Szabó, David Lazer, Kimmo Kaski, János Kertész, and A-L Barabási. Structure and tie strengths in mobile communication networks. *Proceedings of the national academy of sciences*, 104(18):7332–7336, 2007.

OpenReview. OpenReview. `https://openreview.net`, 2026.

Tingyue Pan, Jie Ouyang, Mingyue Cheng, Qingchuan Li, Zirui Liu, Daoyu Wang, Mingfan Pan, Shuo Yu, and Qi Liu. Paperscout: An autonomous agent for academic paper search with process-aware sequence-level policy optimization. *arXiv preprint arXiv:2601.10029*, 2026.

Shraman Pramanick, Rama Chellappa, and Subhashini Venugopalan. Spiqa: A dataset for multimodal question answering on scientific papers. *Advances in Neural Information Processing Systems*, 37: 118807–118833, 2024.

Ori Press, Andreas Hochlehnert, Ameya Prabhu, Vishaal Udandarao, Ofir Press, and Matthias Bethge. Citeme: Can language models accurately cite scientific claims? *Advances in Neural Information Processing Systems*, 37:7847–7877, 2024.

Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: Bm25 and beyond. *Information Retrieval*, 3(4):333–389, 2009.

Chenyang Shao, Yong Li, and Fengli Xu. Scinetbench: A relation-aware benchmark for scientific literature retrieval agents. *arXiv preprint arXiv:2601.03260*, 2025a.

Rulin Shao, Rui Qiao, Varsha Kishore, Niklas Muennighoff, Xi Victoria Lin, Daniela Rus, Bryan Kian Hsiang Low, Sewon Min, Wen tau Yih, Pang Wei Koh, and Luke Zettlemoyer. ReasonIR: Training retrievers for reasoning tasks. In *Second Conference on Language Modeling*, 2025b.

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

Amanpreet Singh, Mike D'Arcy, Arman Cohan, Doug Downey, and Sergey Feldman. Scirepeval: A multi-format benchmark for scientific document representations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 5548–5566, 2023.

Yiwen Song, Yale Song, Tomas Pfister, and Jinsung Yoon. Paperorchestra: A multi-agent framework for automated ai research paper writing. *arXiv preprint arXiv:2604.05018*, 2026.

Karen Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21, 1972.

Jiabin Tang, Lianghao Xia, Zhonghang Li, and Chao Huang. Ai-researcher: Autonomous scientific innovation. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

R Endre Tarjan. A note on finding the bridges of a graph. *Information Processing Letters*, 2(6): 160–161, 1974.

Tongyi DeepResearch Team, Baixuan Li, Bo Zhang, Dingchu Zhang, Fei Huang, Guangyu Li, Guoxin Chen, Huifeng Yin, Jialong Wu, Jingren Zhou, et al. Tongyi deepresearch technical report. *arXiv preprint arXiv:2510.24701*, 2025.

Edan Toledo, Karen Hambardzumyan, Martin Josifoski, RISHI HAZRA, Nicolas Baldwin, Alexis Audran-Reiss, Michael Kuchnik, Despoina Magka, Minqi Jiang, Alisia Maria Lupidi, et al. Ai research agents for machine learning: Search, exploration, and generalization in mle-bench. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

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

Bin Wang, Chao Xu, Xiaomeng Zhao, Linke Ouyang, Fan Wu, Zhiyuan Zhao, Rui Xu, Kaiwen Liu, Yuan Qu, Fukai Shang, et al. Mineru: An open-source solution for precise document content extraction. *arXiv preprint arXiv:2409.18839*, 2024a.

Jianyou Andre Wang, Kaicheng Wang, Xiaoyue Wang, Prudhviraj Naidu, Leon Bergen, and Ramamohan Paturi. Scientific document retrieval using multi-level aspect-based queries. *Advances in Neural Information Processing Systems*, 36:38404–38419, 2023.

Yidong Wang, Qi Guo, Wenjin Yao, Hongbo Zhang, Xin Zhang, Zhen Wu, Meishan Zhang, Xinyu Dai, Min Zhang, Qingsong Wen, et al. Autosurvey: Large language models can automatically write surveys. *Advances in neural information processing systems*, 37:115119–115145, 2024b.

Duncan J Watts and Steven H Strogatz. Collective dynamics of 'small-world'networks. *nature*, 393 (6684):440–442, 1998.

Yixuan Weng, Minjun Zhu, Guangsheng Bao, Hongbo Zhang, Jindong Wang, Yue Zhang, and Linyi Yang. Cycleresearcher: Improving automated research via automated review. In *The Thirteenth International Conference on Learning Representations*, 2025.

Shirley Wu, Shiyu Zhao, Michihiro Yasunaga, Kexin Huang, Kaidi Cao, Qian Huang, Vassilis N Ioannidis, Karthik Subbian, James Zou, and Jure Leskovec. Stark: Benchmarking llm retrieval on textual and relational knowledge bases. *Advances in Neural Information Processing Systems*, 37: 127129–127153, 2024.

Yinuo Xu, Shuo Lu, Jianjie Cheng, Meng Wang, Qianlong Xie, Xingxing Wang, Ran He, and Jian Liang. How to train your deep research agent? prompt, reward, and policy optimization in search-r1. *arXiv preprint arXiv:2602.19526*, 2026.

Zhaorui Yang, Bo Pan, Han Wang, Yiyao Wang, Xingyu Liu, Luoxuan Weng, Yingchaojie Feng, Haozhe Feng, Minfeng Zhu, Bo Zhang, et al. Multimodal deepresearcher: Generating text-chart interleaved reports from scratch with agentic framework. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 34368–34377, 2026.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, et al. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025.

Leqi Zheng, Jiajun Zhang, Canzhi Chen, Chaokun Wang, Hongwei Li, Yuying Li, Yaoxin Mao, Shannan Yan, Zixin Song, Zhiyuan Feng, et al. What should i cite? a rag benchmark for academic citation prediction. In *Proceedings of the ACM Web Conference 2026*, pp. 1852–1863, 2026.

Minjun Zhu, Zhen Lin, Yixuan Weng, Panzhong Lu, Qiujie Xie, Yifan Wei, Sifan Liu, QiYao Sun, and Yue Zhang. Autofigure: Generating and refining publication-ready scientific illustrations. In *The Fourteenth International Conference on Learning Representations*, 2026.

A LIMITATIONS AND FUTURE WORK

LINEAGE is designed as a benchmark for empirical lineage retrieval grounded in empirical comparison baselines. The current release focuses on AI conference papers from 2018 to 2025, which provides a controlled and auditable setting for evaluating empirical lineage retrieval. A natural extension is to include journal papers and publications from other domains. However, such expansion depends on the licensing conditions of the underlying publications and on whether other domains provide mature and openly accessible APIs that support large-scale and reproducible data collection.

The benchmark focuses on paper-level retrieval over lineage graphs. Future work can build on LINEAGE to support more advanced research-agent settings, including interactive lineage exploration, lineage-based question answering, survey generation, and contribution positioning. These extensions would allow researchers to evaluate not only whether a system retrieves empirical lineage papers, but also whether it can use the recovered empirical comparison structure to support scientific reasoning and discovery.

Broader Impact. LINEAGE is intended to support more reliable literature retrieval and research-agent evaluation by distinguishing empirical lineage from topical or citation-based relevance. Its positive impact lies in helping researchers better compare retrieval methods and build research agents. A potential negative impact is that automatically constructed lineage graphs may be over-interpreted as complete or authoritative maps of scientific influence. This risk is mitigated by defining lineage relations narrowly as empirical baseline relations, reporting extraction and calibration procedures, and releasing the benchmark as an evaluation resource rather than a deployed decision system.

B EXTENDED RELATED WORK

Academic research benchmarks evaluate whether AI systems can support specific stages of academic research or complete research workflows. Early benchmarks and corpora provided evaluation resources for scholarly document processing (Lo et al., 2020), citation-aware representation and recommendation (Cohan et al., 2020), and broader scientific document understanding (Singh et al., 2023). Recent benchmarks move toward more realistic scientific retrieval and evidence-grounded evaluation. DORIS-MAE evaluates aspect-based scientific queries (Wang et al., 2023), LitSearch studies complex literature-retrieval questions (Ajith et al., 2024), and STARK and SciNetBench evaluate retrieval over textual and relational scientific knowledge bases (Shao et al., 2025a; Wu et al., 2024). Citation-oriented benchmarks evaluate local citation recommendation, reviewer-identified missing citations, citation attribution, and RAG-based citation prediction (Kang et al., 2018; Long et al., 2024; Press et al., 2024; Zheng et al., 2026), while scientific QA benchmarks measure claim verification, full-paper question answering, and multi-document or multimodal evidence aggregation (Dasigi et al., 2021; Lee et al., 2023; Li et al., 2024; Pramanick et al., 2024; Wadden et al., 2020). Research-agent benchmarks further evaluate broader academic workflows, including report generation, experiment execution, and data analytics (Chen et al., 2025a;c;d; Huang et al., 2024; Jiang et al., 2025; Majumder et al., 2024). Literature retrieval benchmarks evaluate whether agents can retrieve, rank, and synthesize relevant publications for downstream research tasks (Asai et al., 2026; He et al., 2025; Pan et al., 2026). However, existing paper retrieval benchmarks mainly evaluate topical relevance, citation grounding, or evidence support, while overlooking empirical lineage relations among papers.

LLM-based research agents have recently received growing attention for automating the full research lifecycle, including idea generation, data collection, experiment execution, method optimization, iterative trial planning, and research evaluation (Biswas et al., 2026; Chen et al., 2026a; Lu et al., 2026; Lyu et al., 2026; Tang et al., 2025). Recent work on research ideation showed the promise of automated idea generation in terms of feasibility and diversity (Si et al., 2025). Studies on experiment execution further formulated AI research as a search process over runnable solutions, with an emphasis on search strategies, operators, and feedback-based exploration (Toledo et al., 2025; Weng et al., 2025). Beyond ideation and experimentation, research agents have also been developed to support research writing, survey generation, and figure generation (Chen et al., 2025b; Song et al., 2026; Wang et al., 2024b; Yang et al., 2026; Zhu et al., 2026). Literature retrieval agents further supported research workflows by retrieving, ranking, and synthesizing relevant publications for downstream research tasks (Asai et al., 2026; He et al., 2025; Pan et al., 2026). Recent studies investigated how

multi-round retrieval agents can be improved through prompt design, reward modeling, and policy optimization (Xu et al., 2026), highlighting the importance of literature retrieval in research agents.

C Evaluation Settings

Our source code and benchmark dataset are available at the anonymous GitHub repository: `https://anonymous.4open.science/r/Lineage-Benchmark-41F8`. Unless otherwise specified, we use a default random seed of 42. For both local and online LLMs, we use deterministic decoding by setting `temperature=0` and `do_sample=False` to improve reproducibility. Furthermore, the intermediate outputs produced by search engines and LLM-based search agents have been stored and will be released for reproduction.

C.1 Metrics

The retrieval metrics evaluate the overall ability of a retriever to recover the unordered lineage ground-truth papers for each query paper and place them near the top of the ranking, without considering their graph distance or relation type. **Recall@k** quantifies how much of the lineage ground-truth papers are covered by the top-k retrieved papers. **Precision@k** captures the fraction of top-k retrieved papers that belong to the lineage ground-truth papers. **Hit Rate@k** indicates whether at least one lineage ground-truth paper appears in the top-k retrieved results. **nDCG@k** characterizes how well lineage ground-truth papers are ranked near the top, using binary relevance because the ground truth is unordered. **MRR@k** reflects how early the first lineage ground-truth paper appears within the top-k ranking.

We further introduce structural metrics to evaluate whether retrieved lineage papers are structurally close to the query paper in the lineage graph. Given a directed lineage graph $G = (V, E)$, each node $v \in V$ denotes a paper, and each directed edge $e = (v_b, v_s) \in E$ indicates that the baseline paper v_b is a predecessor of the source paper v_s. For a query paper $v_q \in V$, let $Y_q \subseteq V$ denote its lineage ground-truth papers, and let $\mathbf{R}_k(v_q) = (r_1, \ldots, r_k)$ denote the ranked top-k retrieval list. We use $R_k(v_q) = \{r_1, \ldots, r_k\}$ to denote the corresponding retrieved set. Since Y_q may contain both predecessors and successors of v_q, we compute graph distance on the undirected version of G, denoted by G^u. Let $\mathrm{hop}(v_q, v)$ denote the shortest-path distance between v_q and v in G^u.

Hop-Weighted Recall@k. Hop-Weighted Recall@k measures whether the retrieved results cover lineage papers that are structurally close to the query paper. It assigns larger weights to papers with smaller graph distances. Formally, Hop-Weighted Recall@k is defined as,

$$\mathrm{HWR@}k(v_q) = \frac{\sum_{v \in R_k(v_q) \cap Y_q} \frac{1}{\mathrm{hop}(v_q, v)}}{\sum_{v \in Y_q} \frac{1}{\mathrm{hop}(v_q, v)}}.$$

Direct Recall@k. Direct Recall@k focuses on the most immediate lineage relations. It measures the fraction of one-hop ground-truth neighbors that appear in the top-k retrieved results. The one-hop ground-truth set is defined as,

$$Y_q^{(1)} = \{v \in Y_q \mid \mathrm{hop}(v_q, v) = 1\}.$$

Accordingly, Direct Recall@k is computed as,

$$\mathrm{DR@}k(v_q) = \frac{|R_k(v_q) \cap Y_q^{(1)}|}{|Y_q^{(1)}|}.$$

Graph-weighted nDCG@k. Graph-weighted nDCG@k evaluates ranking quality under a graph structure. It uses hop-based graded relevance, so that closer lineage papers receive larger gains when ranked higher. For each retrieved paper r_i, its graph-weighted gain is defined as,

$$\mathrm{gain}(r_i) = \begin{cases} \frac{1}{\mathrm{hop}(v_q, r_i)}, & r_i \in Y_q, \\ 0, & r_i \notin Y_q. \end{cases}$$

The resulting graph-weighted DCG is computed as,

$$\mathrm{GDCG@}k(v_q) = \sum_{i=1}^{k} \frac{\mathrm{gain}(r_i)}{\log_2(i+1)}.$$

Let $(y_1, \ldots, y_{|Y_q|})$ be the papers in Y_q sorted by increasing $\mathrm{hop}(v_q, y_i)$. The ideal graph-weighted DCG is then defined as,

$$\mathrm{IGDCG@}k(v_q) = \sum_{i=1}^{\min(k,|Y_q|)} \frac{\frac{1}{\mathrm{hop}(v_q,y_i)}}{\log_2(i+1)}.$$

Finally, graph-weighted nDCG@k is given by,

$$\mathrm{GN@}k(v_q) = \frac{\mathrm{GDCG@}k(v_q)}{\mathrm{IGDCG@}k(v_q)}.$$

C.2 Retriever

Sparse Retrieval. BM25 indexes the `Lineage` corpus with `bm25s`[2] library. Corpus and query texts are tokenized with the default BM25 scorer. TF-IDF retrieval is implemented with `scikit-learn`[3]. The L2-normalized corpus matrix is scored against query vectors by sparse inner products.

Dense Retrieval. For dense retrievers, we evaluate Contriever, BGE-M3, Jina-Embedding-v5, Qwen3-Embedding, NV-Embedding-v2, and ReasonIR. The corresponding repository links in the HuggingFace are listed in Table 5. For Qwen3-Embedding, we include the 0.6B, 4B, and 8B variants. All methods use cosine similarity as the distance metric.

Agentic Retrieval. We evaluate four agentic retrievers. Tongyi-DeepResearch uses Tongyi-DeepResearch-30B-A3B as the search agent, with Qwen3-Embedding-4B as search backbones. AgentIR uses `gpt-4o-mini`[4] as the search agent and AgentIR-4B as the dense retriever. AgentIR-Qwen3-E keeps the same agent loop but replaces the retriever with Qwen3-Embedding-4B. For PaSa, we leverage PaSa-7B-Crawler and PaSa-7B-Selector on our LINEAGE, replacing its original arXiv-based source.

Search Engines. Google Search is queried through `SerpAPI`[5]. Queries are restricted to academic websites such as arXiv, ACL Anthology, OpenReview, and related venues, and results are limited to 2018–2025. Google Scholar retrieval also uses `SerpAPI`[6] with the same year limitation.

LLM Search. GPT-5.4, Gemini-3.1, and DeepSeek-V3.2 are evaluated through APIs with web search enforced. Each model receives the retrieval query, allowed venues, and the 2018–2025 year constraint, and is required to return a ranked JSON list of paper titles by the prompt in Appendix I.3.

For corpus-based retrieval, each candidate paper is represented only by its title and abstract; full-text content is excluded from retrieval. To ensure a fair top-k evaluation, the query paper itself is removed from the retrieved candidate list, since retrievers may rank it highly based on direct lexical or semantic overlap with the query, even though it is not part of the ground-truth set.

D Additional Results

D.1 Retrieval Results on Lineage-0.8k

Due to API rate limits, the *Search Engine* and *LLM Search* methods are evaluated on a sampled subset, denoted as LINEAGE-0.8k. This subset contains 800 queries, with 100 queries evenly sampled from each query type. For a fair comparison, we also evaluate the corpus-based retrieval methods on the same subset and report the results in Table 6. The results are consistent with the main findings in Table 3, further supporting our conclusions drawn from the full LINEAGE dataset.

[2] `https://github.com/xhluca/bm25s`
[3] `https://scikit-learn.org`
[4] `https://platform.openai.com/docs/models`
[5] `https://serpapi.com/google-search-api`
[6] `https://serpapi.com/google-scholar-api`

Table 5: Model checkpoints on HuggingFace.

Model	Model checkpoints	License
Contriever	https://huggingface.co/facebook/contriever-msmarco	CC-BY-NC-4.0
BGE-M3	https://huggingface.co/BAAI/bge-m3	MIT
Jina-Embedding-v5	https://huggingface.co/jinaai/jina-embeddings-v5-text-small	CC-BY-NC-4.0
Qwen3-Embedding-0.6B	https://huggingface.co/Qwen/Qwen3-Embedding-0.6B	Apache-2.0
Qwen3-Embedding-4B	https://huggingface.co/Qwen/Qwen3-Embedding-4B	Apache-2.0
Qwen3-Embedding-8B	https://huggingface.co/Qwen/Qwen3-Embedding-8B	Apache-2.0
NV-Embedding-v2	https://huggingface.co/nvidia/NV-Embed-v2	CC-BY-NC-4.0
ReasonIR	https://huggingface.co/reasonir/ReasonIR-8B	CC-BY-NC-4.0
Tongyi-DeepResearch-30B-A3B	https://huggingface.co/Alibaba-NLP/Tongyi-DeepResearch-30B-A3B	Apache-2.0
AgentIR-4B	https://huggingface.co/Tevatron/AgentIR-4B	Not specified
PaSa-7B-Crawler	https://huggingface.co/bytedance-research/pasa-7b-crawler	CC-BY-NC-SA-4.0
PaSa-7B-Selector	https://huggingface.co/bytedance-research/pasa-7b-selector	CC-BY-NC-SA-4.0

Table 6: Evaluation results of representative retrieval methods on LINEAGE-0.8k.

Category	Retrieval Method	Retrieval Metric					Structural Metric		
		R@20	*P*@20	*H*@20	*N*@20	*M*@20	*HWR*@20	*DR*@20	*GN*@20
Sparse Retrieval	BM25	23.08	6.59	53.63	18.11	26.79	23.84	25.46	18.19
	TF-IDF	22.17	6.39	54.00	17.07	24.63	22.86	24.43	17.16
Dense Retrieval	Contriever	19.09	5.80	50.50	15.09	22.96	19.71	20.87	15.00
	BGE-M3 (0.6B)	10.84	2.98	34.00	8.53	14.25	11.18	11.85	8.54
	Jina-Embedding-v5 (0.6B)	29.80	8.59	61.12	22.78	30.11	30.80	32.76	22.64
	Qwen3-Embedding (0.6B)	27.70	8.77	60.50	22.01	29.80	28.79	30.79	22.00
	Qwen3-Embedding (4B)	**32.74**	**10.26**	63.25	**25.73**	32.99	**34.12**	**36.77**	**25.62**
	Qwen3-Embedding (8B)	30.38	9.80	61.62	24.39	33.28	31.47	33.33	24.03
	NV-Embedding-v2 (8B)	32.02	9.82	**65.38**	25.25	**34.06**	33.35	36.20	25.20
	ReasonIR (8B)	26.56	7.88	60.50	21.27	30.04	27.84	30.70	21.43
Agentic Retrieval	Pasa	22.86	6.66	54.43	18.98	27.43	23.45	24.80	18.71
	AgentIR	27.61	8.29	60.12	22.15	31.31	28.85	31.43	22.28
	AgentIR-Qwen3-Embedding	28.78	8.82	59.25	22.93	30.98	29.88	31.81	22.82
	Tongyi-DeepResearch	31.87	9.92	63.88	24.99	33.09	33.23	35.89	24.99
Search Engine	Google Search	9.64	0.40	32.37	6.64	10.62	9.99	2.23	1.42
	Google Scholar	7.57	2.00	23.62	5.23	8.11	8.01	8.78	5.14
LLM Search	GPT-5.4	7.24	3.30	29.13	7.12	13.56	7.42	7.87	6.65
	Gemini-3.1	4.47	1.97	21.50	4.58	10.35	4.51	4.46	4.23
	DeepSeek-V3.2	4.52	1.82	18.12	4.55	8.92	4.51	4.59	4.23

D.2 METHOD-LEVEL RESULTS ACROSS LINEAGE HOPS

Table 7 presents Recall@20 for each representative retrieval method across one-hop to seven-hop lineage targets. The method-level results reveal a consistent trend across hops that Recall@20 generally decreases as hop distance increases, showing that longer-hop empirical lineage papers are harder to retrieve than direct empirical comparison neighbors. For the direct one-hop setting, Tongyi-DeepResearch achieves the best Recall@20, followed by Qwen3-E. (4B). When averaging uniformly over all hop columns, Qwen3-E. (8B) performs best, followed by Qwen3-E. (4B).

D.3 METHOD-LEVEL RESULTS ACROSS QUERY TYPES

Tables 8 and 9 report Recall@20 and HWR@20 for each representative retrieval method across the eight query types. The query-type columns use the following abbreviations. ***Title*** denotes the direct title query, and ***Title-Ctx*** denotes the title-context query formed by concatenating the title and abstract. ***Top-KW*** and ***Top-Sem*** refer to topical keyword and topical semantic queries, respectively. ***Ctx-KW*** and ***Ctx-Sem*** denote context keyword and context semantic queries focused on task motivation and application domain, while ***Meth-KW*** and ***Meth-Sem*** denote method keyword and method semantic queries focused on method family, modeling idea, or technical mechanism.

The method-level results reveal the same pattern as illustrated in Table 4. Different retrieval families perform better on different query types, with agentic methods being strong on title, topical semantic, and context keyword queries, while dense retrievers remain competitive on topical keyword, context semantic, and method semantic queries. Across query types, Tongyi-DeepResearch achieves the best

Table 7: Recall@20 results of retrieval methods on LINEAGE across hops.

Category	Retrieval Method	1-*hop*	2-*hop*	3-*hop*	4-*hop*	5-*hop*	6-*hop*	7-*hop*
w/ LINEAGE *Corpus* (Avg.)		28.42	14.02	6.68	3.40	1.74	0.69	0.83
Sparse Retrieval	BM25	25.74	11.93	5.43	2.23	1.26	2.08	1.79
	TF-IDF	23.64	11.53	5.71	2.61	1.43	1.14	0.00
Dense Retrieval	Contriever	21.19	9.88	4.67	2.11	1.08	0.00	0.00
	BGE-M3 (0.6B)	12.34	5.98	3.41	1.74	0.88	0.00	2.68
	Jina-Embedding-v5 (0.6B)	30.46	14.60	6.59	2.97	2.29	0.38	0.00
	Qwen3-Embedding (0.6B)	30.18	15.04	7.13	3.50	2.27	0.00	1.79
	Qwen3-Embedding (4B)	35.09	18.75	9.16	4.24	2.98	0.13	0.00
	Qwen3-Embedding (8B)	33.18	18.79	9.76	5.16	4.18	2.45	0.00
	NV-Embedding-v2 (8B)	34.38	17.44	7.64	3.03	2.31	0.18	1.79
	ReasonIR (8B)	29.11	14.91	7.99	3.44	3.12	1.96	3.57
Agentic Retrieval	Pasa	23.47	11.86	6.06	2.76	1.45	1.30	0.00
	AgentIR$^\diamond$	31.43	13.13	6.01	4.32	0.00	0.00	0.00
	AgentIR-Qwen3-Embedding$^\diamond$	31.81	15.47	6.00	4.15	0.00	0.00	0.00
	Tongyi-DeepResearch$^\diamond$	35.89	16.94	7.93	5.37	1.04	0.00	0.00
w/ Online Sources (Avg.)		6.48	4.61	2.16	1.93	0.05	0.00	0.00
Search Engine	Google Search$^\diamond$	10.54	3.97	0.91	2.64	0.24	0.00	0.00
	Google Scholar$^\diamond$	8.90	2.38	1.75	2.23	0.00	0.00	0.00
LLM Search	GPT-5.4$^\diamond$	7.24	7.87	9.61	5.40	2.27	0.00	0.00
	Gemini-3.1$^\diamond$	4.47	4.46	5.08	2.74	2.04	0.00	0.00
	DeepSeek-V3.2$^\diamond$	4.52	4.59	6.72	1.85	0.45	0.00	0.00

Table 8: Recall@20 results of retrieval methods on LINEAGE across query types. For each method, the best query type is highlighted in **bold**, and the second-best is underlined.

Category	Retrieval Method	Query Type							
		Title	Title-Ctx	Top-KW	Top-Sem	Ctx-KW	Ctx-Sem	Meth-KW	Meth-Sem
w/ LINEAGE *Corpus* (Avg.)		30.36	39.04	29.19	31.07	18.90	26.07	7.46	17.14
Sparse Retrieval	BM25	27.15	**36.85**	26.10	26.58	16.20	21.39	7.34	14.88
	TF-IDF	24.05	**36.31**	23.21	24.25	14.85	20.10	6.72	13.86
Dense Retrieval	Contriever	22.17	**31.71**	21.68	21.92	13.09	19.09	4.96	11.00
	BGE-M3 (0.6B)	10.68	**28.14**	11.72	10.69	6.98	9.30	2.43	5.72
	Jina-Embedding-v5 (0.6B)	31.97	**42.08**	30.40	32.39	19.58	27.94	7.48	18.23
	Qwen3-Embedding (0.6B)	31.68	**40.34**	30.09	32.80	19.48	27.32	7.29	18.67
	Qwen3-Embedding (4B)	37.02	**45.98**	35.38	38.12	23.59	32.07	9.92	23.03
	Qwen3-Embedding (8B)	35.28	**45.41**	33.46	36.21	22.01	29.75	9.82	22.87
	NV-Embedding-v2 (8B)	36.76	**41.42**	34.86	37.73	23.05	31.33	9.55	23.37
	ReasonIR (8B)	29.96	**42.60**	27.94	32.13	17.69	25.20	7.94	19.55
Agentic Retrieval	Pasa	26.81	**30.22**	25.29	27.66	16.60	23.47	7.24	15.76
	AgentIR$^\diamond$	33.63	**40.24**	34.07	37.52	20.99	31.52	6.63	16.31
	AgentIR-Qwen3-Embedding$^\diamond$	37.42	**38.53**	36.87	35.11	25.23	32.67	7.71	16.69
	Tongyi-DeepResearch$^\diamond$	40.43	**46.69**	37.62	41.84	25.19	33.89	9.34	19.99
w/ Online Sources (Avg.)		6.06	3.76	8.96	7.56	5.85	6.50	3.06	5.02
Search Engine	Google Search$^\diamond$	15.51	2.04	**17.83**	14.04	6.37	12.94	2.17	6.19
	Google Scholar$^\diamond$	1.02	0.00	**14.80**	12.78	10.52	7.00	5.84	8.57
LLM Search	GPT-5.4$^\diamond$	7.55	**8.63**	8.05	6.94	6.76	6.81	5.21	7.97
	Gemini-3.1$^\diamond$	6.20	**6.85**	5.53	3.92	4.91	4.08	1.89	2.35
	DeepSeek-V3.2$^\diamond$	6.44	**6.89**	4.87	4.08	4.87	4.27	2.06	2.67

average Recall@20 and HWR@20, followed by Qwen3-E. (4B), while AgentIR-Qwen3-E performs best on title-context queries and NV-E.-v2 performs best on context-semantic queries.

E LINEAGE-HA

To validate the proposed paper type classifier and dual-source lineage extractor, we construct a human-annotated validation set termed LINEAGE-HA. It contains paper-type annotations for 230 papers and lineage-relation annotations for 150 methodology papers. The lineage-relation annotations

Table 9: HWR@20 results of representative retrieval methods on LINEAGE across query types.

Category	Retrieval Method	Query Type							
		Title	Title-Ctx	Top-KW	Top-Sem	Ctx-KW	Ctx-Sem	Meth-KW	Meth-Sem
w/ LINEAGE *Corpus* (Avg.)		31.89	**41.19**	30.63	32.67	19.57	27.06	7.80	18.01
Sparse Retrieval	BM25	28.66	**39.11**	27.51	28.09	16.94	22.42	7.75	15.74
	TF-IDF	25.27	**38.47**	24.39	25.56	15.44	21.00	7.08	14.61
Dense Retrieval	Contriever	23.45	**33.60**	22.84	23.19	13.59	19.91	5.23	11.64
	BGE-M3 (0.6B)	11.24	**29.82**	12.31	11.22	7.20	9.58	2.55	5.98
	Jina-Embedding-v5 (0.6B)	33.73	**44.52**	32.05	34.21	20.38	29.17	7.89	19.20
	Qwen3-Embedding (0.6B)	33.42	**42.65**	31.72	34.66	20.36	28.59	7.69	19.64
	Qwen3-Embedding (4B)	38.81	**48.45**	37.11	40.03	24.52	33.43	10.37	24.11
	Qwen3-Embedding (8B)	36.80	**47.74**	34.92	37.82	22.73	30.80	10.19	23.81
	NV-Embedding-v2 (8B)	38.78	**43.76**	36.69	39.78	23.98	32.67	10.01	24.60
	ReasonIR (8B)	31.44	**45.20**	29.24	33.77	18.24	26.07	8.30	20.56
Agentic Retrieval	Pasa	28.22	**31.88**	26.57	29.13	17.27	24.49	7.62	16.59
	AgentIR$^\diamond$	35.20	**42.59**	35.45	39.31	21.74	32.72	6.82	16.95
	AgentIR-Qwen3-Embedding$^\diamond$	38.98	**40.11**	38.51	36.64	25.91	33.47	7.88	17.55
	Tongyi-DeepResearch$^\diamond$	42.40	**48.78**	39.46	44.01	25.67	34.58	9.83	21.11
w/ Online Sources (Avg.)		6.27	3.94	**9.40**	8.00	6.02	6.54	3.16	5.28
Search Engine	Google Search$^\diamond$	15.98	2.27	**18.76**	14.47	6.61	12.91	2.34	6.60
	Google Scholar$^\diamond$	1.04	0.00	**15.95**	13.85	10.68	7.18	6.11	9.24
LLM Search	GPT-5.4$^\diamond$	7.80	**9.04**	8.09	7.37	6.76	6.83	5.30	8.18
	Gemini-3.1$^\diamond$	6.33	**6.94**	5.49	4.00	5.06	4.02	1.82	2.42
	DeepSeek-V3.2$^\diamond$	6.56	**6.87**	4.82	4.11	5.00	4.21	1.92	2.63

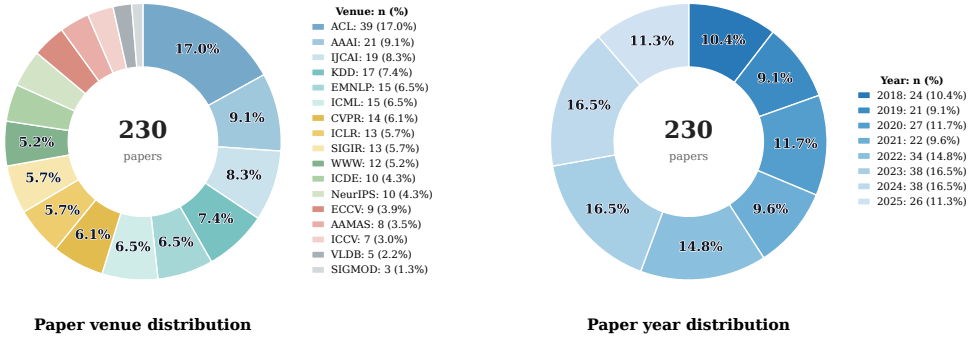


Figure 7: The venue and year distribution of the 230 papers annotated for paper type in LINEAGE-HA.

contain 970 positive lineage relations, with an average of 6.47 relations per paper. The annotated papers cover different publication years, venues, and research areas, as illustrated in Figure 7.

E.1 HUMAN ANNOTATION PROTOCOL

The annotations were produced by recruited independent annotators who are PhD students working in research areas related to the annotated papers. Each annotator was compensated at a rate of USD 10 per hour.

For paper-type annotation, annotators assign each paper to its primary contribution. A paper is labeled as *methodology* if its main contribution is a technical method, as *benchmark* if it mainly contributes an evaluation resource or protocol, and as *survey* if it mainly synthesizes prior work.

For lineage-relation annotation, annotators inspect the full paper, including experimental text, tables, figures, captions, and appendices. They identify methods explicitly used as empirical comparison baselines, map method names and aliases to bibliography entries, and exclude citations that are not used for empirical comparison. Each paper is independently annotated by two annotators. Disagreements and uncertain cases are reviewed before determining the final annotations.

E.2 ANNOTATION AGREEMENT

To assess the reliability of the lineage-relation annotations, we conduct an agreement analysis on a 50-paper subset of LINEAGE-HA. One author independently annotates the same papers using

Table 10: Annotation agreement on a 50-paper subset of LINEAGE-HA.

Metric	Annotation Sets	Score
Krippendorff's α	Annotator 1, Annotator 2, Author	0.8490
Micro Dice	Annotator 1–Annotator 2	0.9757
Micro Dice	Annotator 1–Author	0.9706
Micro Dice	Annotator 2–Author	0.9805

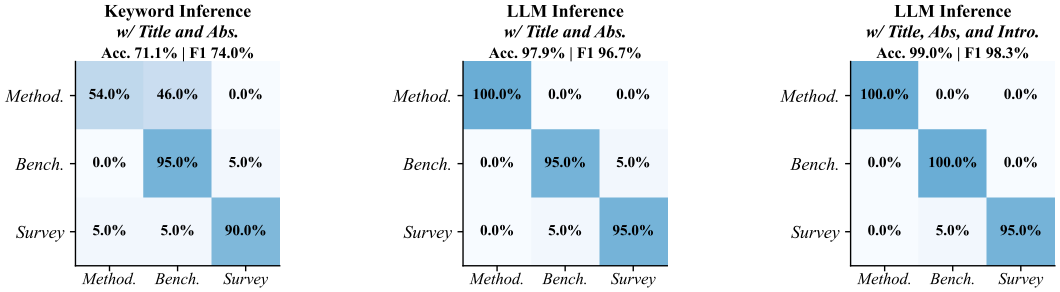


Figure 8: Comparison of paper-type inference strategies on the 230-paper paper-type annotation set in LINEAGE-HA. Each confusion matrix reports predictions over methodology, benchmark, and survey papers, with overall accuracy and macro-F1 shown above each matrix.

the annotation protocol in Section E.1, without access to the labels produced by the independent annotators. The author annotations are used only to assess annotation quality and are not used to construct the final annotations or evaluate the lineage extractor.

For each paper, we compare the sets of lineage relations produced by the two independent annotators and the author. We use Krippendorff's alpha with MASI distance to measure agreement among the three set-valued annotations. We additionally report pairwise micro-averaged Sørensen–Dice scores. Agreement is computed only from the three independently produced annotation sets, while the adjudicated final annotations are excluded.

As shown in Table 10, Krippendorff's alpha reaches 0.8490, while all pairwise Micro-Dice scores exceed 0.97. These results indicate that the annotations produced by the independent annotators are highly consistent with each other and with annotations independently produced by an author under the same evidence-based protocol.

E.3 PAPER TYPE CLASSIFICATION TASK

We sample 230 papers across venues and publication years for paper-type annotation, including 150 methodology papers, 40 benchmark papers, and 40 survey papers, as illustrated in Figure 7. Based on this annotated set, we evaluate our paper type classifier against several alternatives, including:

- **Keyword Classifier.** This classifier serves as a lightweight heuristic baseline, relying on keyword matching to distinguish survey, benchmark, and methodology papers.
- **LLM *w/* Title & Abstract (Used in our pipeline).** This method uses an LLM to infer the primary contribution type from the paper title and abstract, balancing semantic understanding with minimal input cost.
- **LLM *w/* Title & Abstract & Introduction.** This variant provides the LLM with the introduction in addition to the title and abstract, enabling more informed classification.

Figure 8 compares three paper-type inference strategies on the 230-paper paper-type annotation set in LINEAGE-HA. The keyword-based classifier performs substantially worse than the LLM-based classifiers, particularly on methodology papers. In contrast, the LLM classifier using only titles and abstracts achieves strong performance across methodology, benchmark, and survey papers. Adding the introduction provides only a marginal improvement while increasing input length and parsing cost. We therefore use the title-and-abstract LLM classifier in our LINEAGE construction pipeline.

Table 11: Lineage extraction results of our dual-source lineage extractor and its variants on LINEAGE-HA with different input sources.

Extractor	Source	Recall	Precision	micro F1
w/ Integration	MinerU	<u>94.24</u>	83.71	88.66
	GROBID	85.25	81.87	83.52
	Dual	93.71	**92.21**	**92.95**
w/ Verifier	MinerU	93.53	<u>92.04</u>	<u>92.77</u>
	GROBID	80.22	86.27	83.13
	Dual	**95.50**	84.15	89.47
w/ Filtering	MinerU	75.18	88.00	81.09
	GROBID	65.83	76.09	70.59
	Dual	69.99	86.62	77.26
w/ Evidence	MinerU	85.79	87.85	86.81
	GROBID	67.99	78.10	72.69
	Dual	85.79	89.66	87.68

E.4 LINEAGE RELATION EXTRACTION TASK

The lineage-relation annotation set contains 150 methodology papers and 970 positive lineage relations, corresponding to an average of 6.47 relations per paper. We evaluate our dual-source lineage extractor and its alternatives on LINEAGE-HA.

The extraction methods include:

- **Extractor *w/* Integration (Used in our pipeline).** This method performs extraction over both MinerU- and GROBID-parsed full texts and integrates their outputs through an LLM-based resolution step. The dual-source integration leverages complementary parsing results to improve coverage and robustness against parser-specific errors.

- **Extractor *w/* Verifier.** These variants first extract candidate lineage relations and then ask the LLM to verify them against the paper content, aiming to reduce unsupported baseline predictions.

- **Extractor *w/* Filtering.** These variants introduce an additional filtering stage to locate likely experimental comparison sections before performing lineage extraction.

- **Extractor *w/* Evidence.** These methods extract lineage relations from selected evidence units and reference entries, focusing the LLM on comparison-relevant content rather than the entire paper.

Table 11 reports the lineage extraction performance of different extractor variants with MinerU, GROBID, and their dual-source combination. Among all variants, the dual-source extractor with integration achieves the best micro-F1 score of 92.95, with a recall of 93.71 and the highest precision of 92.21. Although the dual-source verifier obtains the highest recall of 95.50, its lower precision indicates that additional verification does not sufficiently suppress unsupported lineage candidates. The filtering and evidence-based variants perform worse overall, suggesting that empirical baseline evidence is not always confined to a small set of pre-selected sections or evidence units.

The comparison across input sources further shows that MinerU and GROBID provide complementary paper representations. For the integration-based extractor, MinerU alone achieves high recall but substantially lower precision, while GROBID alone yields lower overall performance. Combining the two parsing sources improves precision while preserving high recall, resulting in the strongest overall micro-F1. We therefore adopt the dual-source extractor with integration in the LINEAGE-Constructor pipeline.

F CONSTRUCTION DETAILS

F.1 METADATA AND PDF ACQUISITION

Acquisition Sources. We acquire paper PDFs through a venue-aware collection pipeline that integrates 11 open-access sources. Starting from a canonical venue-year paper list, the collector queries venue-specific PDF sources in a fixed priority order, as shown in Table 12. Official proceedings are used as the primary source when available, while OpenReview and arXiv serve as complementary fallback sources for venues with incomplete official coverage or restricted automated access. After a PDF is obtained, the file is validated before downstream parsing, and abnormal or incomplete files are filtered to reduce noise in lineage extraction.

Table 12: Venue-specific sources for metadata supplementation and PDF acquisition. PDF sources are acquired from left to right.

Venue	PDF Acquisition Order	Official OA Source
AAAI	AAAI OJS → arXiv	AAAI OJS
AAMAS	IFAAMAS → OpenReview → arXiv	IFAAMAS
ACL	ACL Anthology → arXiv	ACL Anthology
ECCV	ECVA → arXiv	ECVA
CVPR	CVF → OpenReview → arXiv	CVF
EMNLP	ACL Anthology → arXiv	ACL Anthology
ICCV	CVF → arXiv	CVF
ICDE	arXiv	–
ICLR	OpenReview → arXiv	OpenReview
ICML	PMLR → OpenReview → arXiv	PMLR
IJCAI	IJCAI → arXiv	IJCAI
KDD†	OpenReview → arXiv	–
NeurIPS	NeurIPS → OpenReview → arXiv	NeurIPS
SIGIR†	arXiv	–
SIGMOD†	arXiv	–
VLDB	PVLDB → arXiv	PVLDB
WWW†	OpenReview → arXiv	–

†ACM proceedings are openly accessible through the ACM Digital Library, but it does not provide a public bulk-download API and restricts programmatic access. We therefore do not use it as an automated PDF source.

F.2 QUALITY FILTERING AND CALIBRATION

PDF Filtering. After metadata enrichment and PDF validation, we apply two lightweight diagnostic filters to reduce noisy records before downstream parsing and lineage extraction. First, we remove PDFs with fewer than six pages. This threshold serves as a conservative sanity check, since regular research papers in the venues covered by LINEAGE are typically longer, whereas shorter files are more likely to be extended abstract-like non-regular-track records collected from heterogeneous open-access sources. It reduces the possibility that incomplete or irrelevant documents enter the full-paper extraction stage.

Abstract Calibration. We inspect abstracts with abnormal surface forms, including abstracts shorter than 200 characters, longer than 2000 characters, or lacking terminal punctuation. These cases are likely to reflect metadata inconsistencies rather than genuine abstract content, such as truncated abstracts, duplicated boilerplate text, or incorrectly captured introduction paragraphs. We manually review and correct these records to maintain a consistent metadata representation for query generation and lineage construction.

Graph Calibration. Table 13 reports the stage-wise effect of lineage graph calibration, where we employ two local graph diagnostics to demonstrate the properties of graph structures. The clustering coefficient (Watts & Strogatz, 1998) measures local neighborhood closure, namely how often nodes

Table 13: Stage-wise effect of lineage graph calibration.

Stage	Node	Edge	Clustering	Edge Jaccard
Raw Lineage Graph	89,485	238,511	0.1627	0.1022
Temporal Calibration	89,441	238,075	0.1635	0.1020
Structural Calibration	67,502	164,948	0.3035	0.1420

sharing a neighbor are also connected. The edge-neighborhood Jaccard similarity (Onnela et al., 2007) measures the overlap between the neighbor sets of two nodes connected by a lineage edge.

Temporal calibration removes 436 future-year and cycle-forming edges, accounting for 0.15% of the raw matched edges. It mainly enforces the temporal semantics of lineage relations, where baseline papers should not appear after their corresponding source papers. Structural calibration then corrects abnormal connectivity caused by widely reused baseline methods, which can link otherwise distinct method families and make multi-hop empirical comparison neighborhoods overly broad. It restricts upper-tail connectivity and yields a calibrated graph with 67,502 papers and 164,948 lineage edges. The clustering coefficient increases to 0.3035, and the mean edge-neighborhood Jaccard similarity increases to 0.1420, suggesting stronger local structural coherence after calibration.

F.3 Cost Comparison

We estimate the annotation cost of our automatic labeling system based on two tasks. In our LINEAGE-HA labeling, paper type classification takes approximately 20–60 seconds per paper. For 121,048 papers, this requires around 672–2,017 hours. Lineage relation extraction takes approximately 3-10 minutes per paper. For 112,996 papers, this requires around 5,650–18,833 hours. With a compensation rate of 10 USD per hour, manually completing both tasks would cost around 63.2K–208.5K USD.

In comparison, our automatic system takes only 0.8–1.2 seconds per paper for paper type classification and 15–28 seconds per paper for lineage relation extraction. This corresponds to approximately 497.7–919.2 GPU hours on a single Nvidia H20 GPU. Accounting for GPU depreciation and electricity, the total computational cost is around 228–491 USD. Even using cloud GPU services for estimation, the cost remains below 0.7K USD. This indicates that the automatic pipeline reduces annotation cost by roughly two orders of magnitude.

G Lineage Statistics

In this section, we provide additional statistics for LINEAGE, complementing the corpus-scale summary, lineage graph analysis, and query distribution.

G.1 Corpus Coverage and Source Composition

Figure 9(a) reports the venue-level PDF acquisition coverage. Across 131,179 collected paper records, 121,048 papers have valid PDFs, while 10,131 papers remain without PDFs, giving an overall PDF coverage of 92.3%. The largest venues, including NeurIPS, CVPR, AAAI, and ICML, contribute most of the collected papers and also account for a large portion of the acquired PDFs. Figure 9(b) summarizes the successful PDF sources, with the source-level counts summing to 121,048. ACL Anthology, CVF, NeurIPS, AAAI OJS, PMLR, and OpenReview are the main contributors, while arXiv and smaller venue-specific sources recover additional papers that are not covered by the primary sources.

G.2 Lineage Graph Structure

We further analyze the structural properties of the constructed lineage graphs. Figures 10(a) and (b) show that LINEAGE contains many small method families together with a smaller number of large components. This pattern is expected in AI literature, where most methods are compared within relatively local research areas, while widely used methods can connect otherwise distinct empirical comparison neighborhoods. The degree distribution in Figure 10(c) also exhibits a long-tailed pattern,

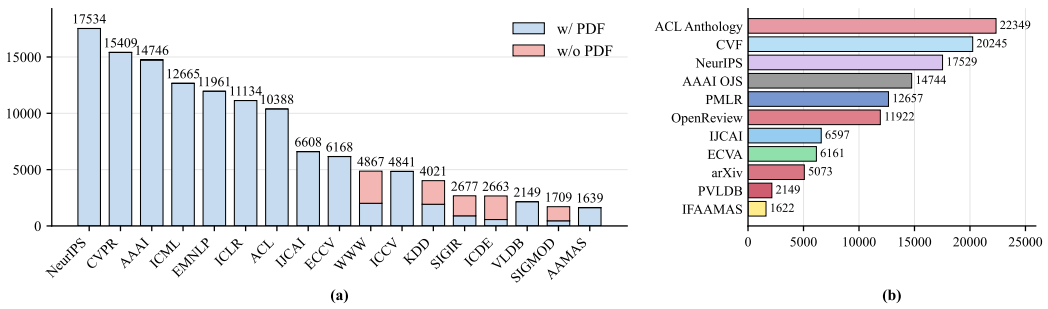


Figure 9: PDF acquisition coverage and source composition. **(a)** Venue-level distribution of papers with and without open-access PDFs. **(b)** Source-level distribution of successfully acquired PDFs.

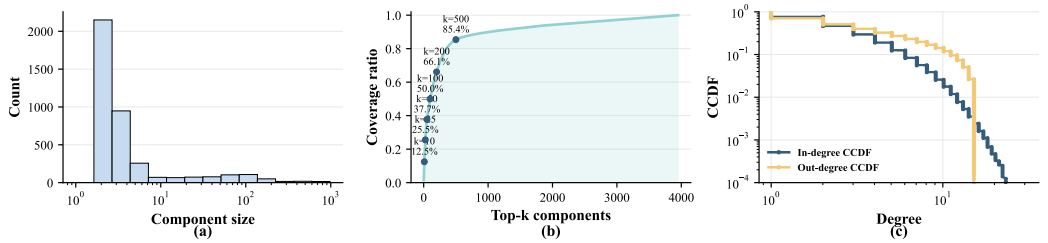


Figure 10: Structural properties of the constructed lineage graphs. **(a)** Distribution of component sizes. **(b)** Node coverage by the largest components. **(c)** In-degree and out-degree distributions.

suggesting that a small number of papers serve as common empirical baselines or later comparison anchors, whereas most papers have a limited number of direct empirical comparison neighbors.

G.3 Query and Ground-truth Lineage Distribution

Figure 11 provides an additional temporal analysis for multi-hop lineage relations. Ancestor papers are on average earlier than the query paper, while successor papers are on average later. The temporal gap also grows with graph distance, indicating that longer-hop lineage targets tend to lie at more distant positions along empirical comparison chains rather than reflect arbitrary graph connections.

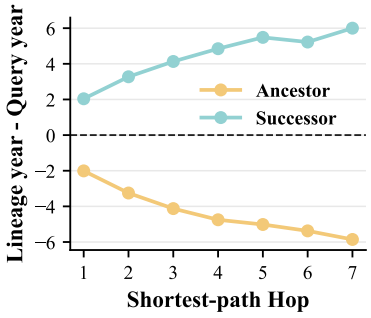


Figure 11: Average publication-year offsets over lineage distance.

Finally, we examine the venue distribution of benchmark queries and their lineage ground-truth papers. As shown in Figure 12, the query set covers all included venues and broadly follows the venue composition of the constructed corpus. The lineage ground-truth distribution is not identical to the query distribution, since each query paper may connect to prior or later papers from different venues. This cross-venue pattern reflects the fact that empirical lineage is not confined within a single publication venue and supports evaluating retrieval across the full LINEAGE corpus.

H Cases

This section provides qualitative examples of LINEAGE to clarify the benchmark semantics and usage. We illustrate how empirical baseline evidence induces lineage graphs, and how multi-facet queries express different search intents for the same query paper.

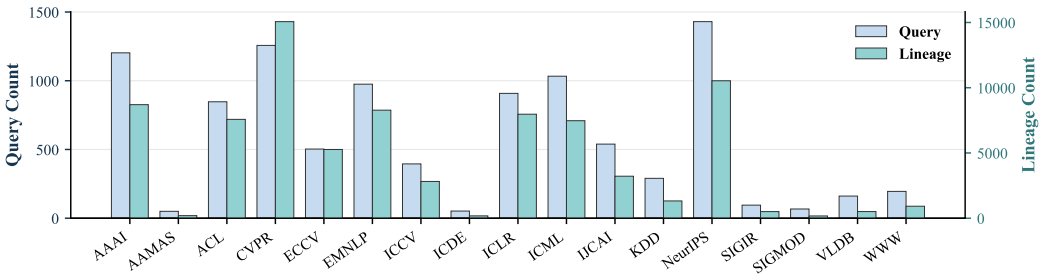


Figure 12: Venue coverage of benchmark queries and lineage ground truth.

H.1 Lineage Graph

We provide two representative lineage subgraphs in Figure 13 and Figure 14 to illustrate the structure of method evolution captured by our benchmark. Each node denotes a paper, and each directed edge indicates that the paper is used as a baseline by the query paper. Benchmark query papers are highlighted in the graphs. Table 14 and Table 15 list the full title of each paper node.

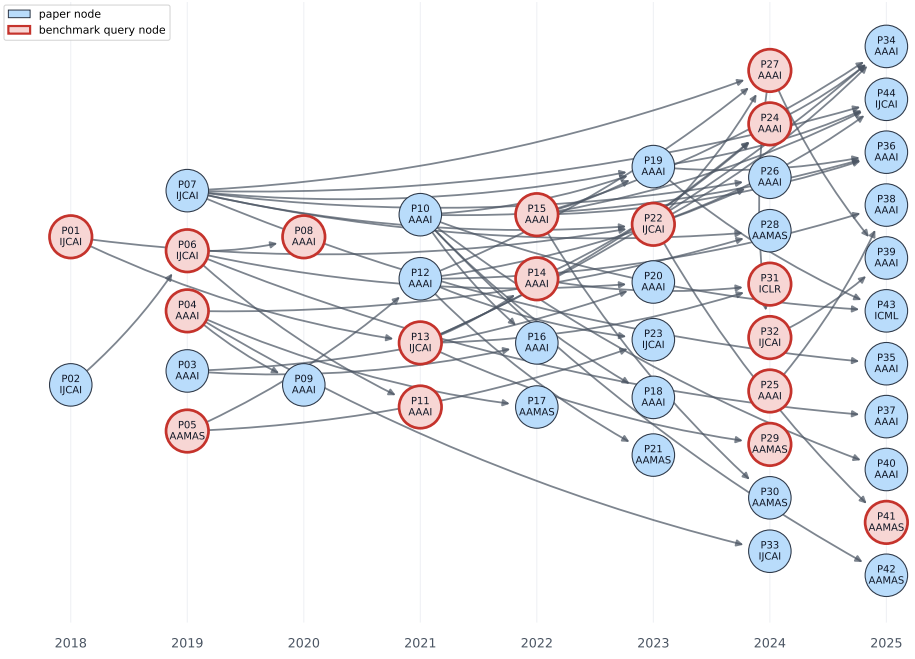


Figure 13: The subgraph of papers in *Multi-Agent Path Finding*.

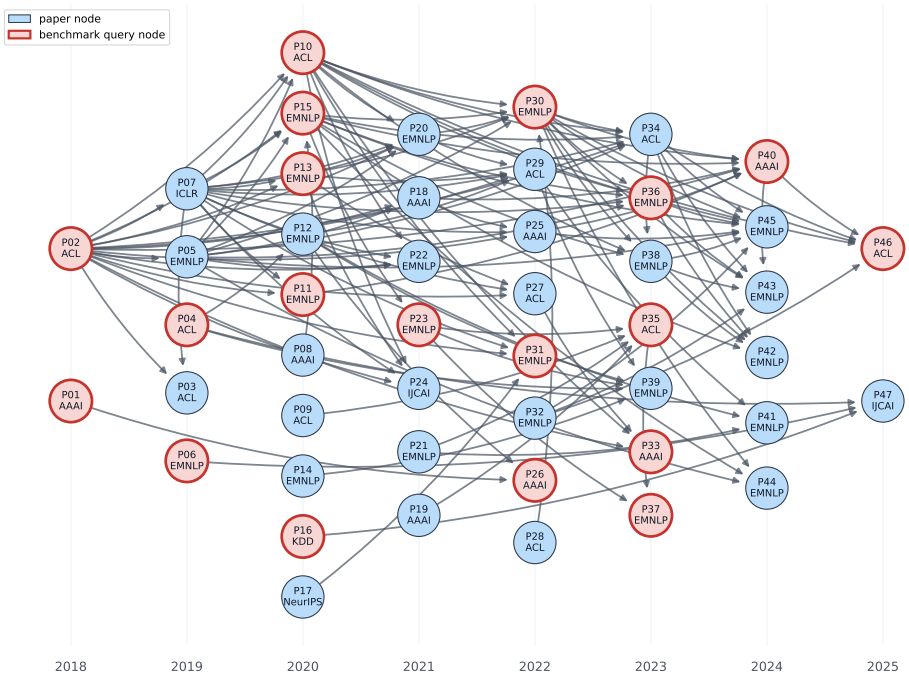


Figure 14: The subgraph of papers in *Task-Oriented Dialogue Systems*.

Table 14: Papers in the subgraph of multi-agent path finding.

Point	Title
P01	Anytime Focal Search with Applications
P02	Extended Increasing Cost Tree Search for Non-Unit Cost Domains
P03	Online Multi-Agent Pathfinding
P04	Searching with Consistent Prioritization for Multi-Agent Path Finding
P05	Task and Path Planning for Multi-Agent Pickup and Delivery
P06	Multi-Agent Pathfinding with Continuous Time
P07	Priority Inheritance with Backtracking for Iterative Multi-agent Path Finding
P08	Generalized and Sub-Optimal Bipartite Constraints for Conflict-Based Search
P09	Idle Time Optimization for Target Assignment and Path Finding in Sortation Centers
P10	EECBS: A Bounded-Suboptimal Search for Multi-Agent Path Finding
P11	Improving Continuous-time Conflict Based Search
P12	Lifelong Multi-Agent Path Finding in Large-Scale Warehouses
P13	Anytime Multi-Agent Path Finding via Large Neighborhood Search
P14	Anytime Multi-Agent Path Finding via Machine Learning-Guided Large Neighborhood Search
P15	MAPF-LNS2: Fast Repairing for Multi-Agent Path Finding via Large Neighborhood Search
P16	Shard Systems: Scalable, Robust and Persistent Multi-Agent Path Finding with Performance Guarantees
P17	Multi-Agent Path Finding for Precedence-Constrained Goal Sequences
P18	Effective Integration of Weighted Cost-to-Go and Conflict Heuristic within Suboptimal CBS
P19	LaCAM: Search-Based Algorithm for Quick Multi-Agent Pathfinding
P20	Periodic Multi-Agent Path Planning
P21	Distributed Planning with Asynchronous Execution with Local Navigation for Multi-agent Pickup and Delivery Problem
P22	Improving LaCAM for Scalable Eventually Optimal Multi-Agent Pathfinding
P23	Multi-Robot Coordination and Layout Design for Automated Warehousing
P24	Adaptive Anytime Multi-Agent Path Finding Using Bandit-Based Large Neighborhood Search
P25	Decentralized Monte Carlo Tree Search for Partially Observable Multi-Agent Pathfinding
P26	Learn to Follow: Decentralized Lifelong Multi-Agent Pathfinding via Planning and Learning
P27	Traffic Flow Optimisation for Lifelong Multi-Agent Path Finding
P28	Attention-based Priority Learning for Limited Time Multi-Agent Path Finding
P29	Confidence-Based Curriculum Learning for Multi-Agent Path Finding
P30	Engineering LaCAM*: Towards Real-time, Large-scale, and Near-optimal Multi-agent Pathfinding
P31	Neural Neighborhood Search for Multi-agent Path Finding
P32	Guidance Graph Optimization for Lifelong Multi-Agent Path Finding
P33	Scalable Mechanism Design for Multi-Agent Path Finding
P34	Anytime Multi-Agent Path Finding with an Adaptive Delay-Based Heuristic
P35	Concurrent Planning and Execution in Lifelong Multi-Agent Path Finding with Delay Probabilities
P36	LNS2+RL: Combining Multi-agent Reinforcement Learning with Large Neighborhood Search in Multi-agent Path Finding
P37	Loosely Synchronized Rule-Based Planning for Multi-Agent Path Finding with Asynchronous Actions
P38	MAPF-GPT: Imitation Learning for Multi-Agent Pathfinding at Scale
P39	Online Guidance Graph Optimization for Lifelong Multi-Agent Path Finding
P40	Windowed MAPF with Completeness Guarantees
P41	Minimizing Makespan with Conflict-Based Search for Optimal Multi-Agent Path Finding
P42	Single-Agent Planning in a Multi-Agent System: A Unified Framework for Type-Based Planners
P43	Hybrid Quantum-Classical Multi-Agent Pathfinding
P44	Multi-Agent Corridor Generating Algorithm

Table 15: Papers in the subgraph of task-oriented dialogue systems.

Point	Title
P01	Augmenting End-to-End Dialogue Systems With Commonsense Knowledge
P02	Mem2Seq: Effectively Incorporating Knowledge Bases into End-to-End Task-Oriented Dialog Systems
P03	A Working Memory Model for Task-oriented Dialog Response Generation
P04	OpenDialKG: Explainable Conversational Reasoning with Attention-based Walks over Knowledge Graphs
P05	Entity-Consistent End-to-end Task-Oriented Dialogue System with KB Retriever
P06	Towards Knowledge-Based Recommender Dialog System
P07	Global-to-local Memory Pointer Networks for Task-Oriented Dialogue
P08	Task-Oriented Dialog Systems That Consider Multiple Appropriate Responses under the Same Context
P09	BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension
P10	Dynamic Fusion Network for Multi-Domain End-to-end Task-Oriented Dialog
P11	Amalgamating Knowledge from Two Teachers for Task-oriented Dialogue System with Adversarial Training
P12	AttnIO: Knowledge Graph Exploration with In-and-Out Attention Flow for Knowledge-Grounded Dialogue
P13	GraphDialog: Integrating Graph Knowledge into End-to-End Task-Oriented Dialogue Systems
P14	Knowledge-Grounded Dialogue Generation with Pre-trained Language Models
P15	Learning Knowledge Bases with Parameters for Task-Oriented Dialogue Systems
P16	Improving Conversational Recommender Systems via Knowledge Graph based Semantic Fusion
P17	A Simple Language Model for Task-Oriented Dialogue
P18	Exploring Auxiliary Reasoning Tasks for Task-oriented Dialog Systems with Meta Cooperative Learning
P19	KG-BART: Knowledge Graph-Augmented BART for Generative Commonsense Reasoning
P20	Contextualize Knowledge Bases with Transformer for End-to-end Task-Oriented Dialogue Systems
P21	EARL: Informative Knowledge-Grounded Conversation Generation with Entity-Agnostic Representation Learning
P22	Intention Reasoning Network for Multi-Domain End-to-end Task-Oriented Dialogue
P23	Neural Path Hunter: Reducing Hallucination in Dialogue Systems via Path Grounding
P24	UniMF: A Unified Framework to Incorporate Multimodal Knowledge Bases intoEnd-to-End Task-Oriented Dialogue Systems
P25	GraphMemDialog: Optimizing End-to-End Task-Oriented Dialog Systems Using Graph Memory Networks
P26	HiTKG: Towards Goal-Oriented Conversations via Multi-Hierarchy Learning
P27	An Interpretable Neuro-Symbolic Reasoning Framework for Task-Oriented Dialogue Generation
P28	Multi-Task Pre-Training for Plug-and-Play Task-Oriented Dialogue System
P29	Towards Large-Scale Interpretable Knowledge Graph Reasoning for Dialogue Systems
P30	Q-TOD: A Query-driven Task-oriented Dialogue System
P31	Robust Task-Oriented Dialogue Generation with Contrastive Pre-training and Adversarial Filtering
P32	UnifiedSKG: Unifying and Multi-Tasking Structured Knowledge Grounding with Text-to-Text Language Models
P33	Towards Complex Scenarios: Building End-to-End Task-Oriented Dialogue System across Multiple Knowledge Bases
P34	Multi-Grained Knowledge Retrieval for End-to-End Task-Oriented Dialog
P35	RHO: Reducing Hallucination in Open-domain Dialogues with Knowledge Grounding
P36	Dual-Feedback Knowledge Retrieval for Task-Oriented Dialogue Systems
P37	Exploring In-Context Learning for Knowledge Grounded Dialog Generation
P38	Retrieval-Generation Alignment for End-to-End Task-Oriented Dialogue System
P39	Well Begun is Half Done: Generator-agnostic Knowledge Pre-Selection for Knowledge-Grounded Dialogue
P40	From Retrieval to Generation: A Simple and Unified Generative Model for End-to-End Task-Oriented Dialogue
P41	Generative Subgraph Retrieval for Knowledge Graph-Grounded Dialog Generation
P42	Learning to Match Representations is Better for End-to-End Task-Oriented Dialog System
P43	Relevance Is a Guiding Light: Relevance-aware Adaptive Learning for End-to-end Task-oriented Dialogue System
P44	Synergizing In-context Learning with Hints for End-to-end Task-oriented Dialog Systems
P45	What are the Generator Preferences for End-to-end Task-Oriented Dialog System?
P46	LLM-Enhanced Query Generation and Retrieval Preservation for Task-Oriented Dialogue
P47	Do Mentioned Items Truly Matter? Enhancing Conversational Recommender Systems with Causal Intervention and Large Language Models

H.2 MULTI-FACET QUERIES

We provide two query cases (sample_id lineage_000234 and lineage_001813) in Table 16 and Table 17 to illustrate how multi-facet queries describe the same query paper from paper-centric, topical, contextual, and methodological perspectives.

Table 16: Selected query example (ID: *lineage_000234*).

Key	Value
Title	SketchGen: Generating Constrained CAD Sketches
Venue	NeurIPS
Year	2021
Abstract	Computer-aided design (CAD) is the most widely used modeling approach for technical design. The typical starting point in these designs is 2D sketches which can later be extruded and combined to obtain complex three-dimensional assemblies. Such sketches are typically composed of parametric primitives, such as points, lines, and circular arcs, augmented with geometric constraints linking the primitives, such as coincidence, parallelism, or orthogonality. Sketches can be represented as graphs, with the primitives as nodes and the constraints as edges. Training a model to automatically generate CAD sketches can enable several novel workflows, but is challenging due to the complexity of the graphs and the heterogeneity of the primitives and constraints ... [total 240 words]
Title-Ctx	SketchGen: Generating Constrained CAD Sketches. + <abstract>.
Top-KW	CAD sketch generation transformer.
Top-Sem	Generative models for CAD sketch creation using transformer architectures and constraint-aware sequences.
Ctx-KW	Parametric CAD design constraint-based modeling.
Ctx-Sem	CAD sketch generation in technical design with parametric primitives and geometric constraints.
Meth-KW	Transformer graph-based sketch representation.
Meth-Sem	Transformer-based generative models for sketch graphs with constraint and primitive type encoding.

Table 17: Selected query example (ID: *lineage_001813*).

Key	Value
Title	HUMUS-Net: Hybrid Unrolled Multi-scale Network Architecture for Accelerated MRI Reconstruction
Venue	NeurIPS
Year	2022
Abstract	In accelerated MRI reconstruction, the anatomy of a patient is recovered from a set of undersampled and noisy measurements. Deep learning approaches have been proven to be successful in solving this ill-posed inverse problem and are capable of producing very high quality reconstructions. However, current architectures heavily rely on convolutions, that are content-independent and have difficulties modeling long-range dependencies in images. Recently, Transformers, the workhorse of contemporary natural language processing, have emerged as powerful building blocks for a multitude of vision tasks. These models split input images into non-overlapping patches, embed the patches into lower-dimensional tokens and utilize a self-attention mechanism that does not suffer from the aforementioned weaknesses of convolutional architectures ... [total 268 words]
Title-Ctx	HUMUS-Net: Hybrid Unrolled Multi-scale Network Architecture for Accelerated MRI Reconstruction. + <abstract>.
Top-KW	MRI reconstruction unrolled network multi-scale.
Top-Sem	Hybrid convolution-transformer networks for MRI reconstruction with multi-scale feature refinement.
Ctx-KW	Accelerated MRI undersampled measurements noise recovery.
Ctx-Sem	MRI reconstruction from undersampled noisy measurements in clinical imaging settings.
Meth-KW	Hybrid convolution transformer multi-scale feature extraction.
Meth-Sem	Unrolled multi-scale networks combining convolutional and transformer blocks for feature refinement.

I PROMPTS

This section presents the prompts used in the LLM-based components of LINEAGE-Constructor and LINEAGE benchmark construction. These prompts cover paper type classification, dual-source lineage extraction, conflict resolution over MinerU and GROBID candidates, and multi-facet retrieval query generation. Each prompt defines the required input fields, decision rules, and JSON output format, making the automatic construction process reproducible and traceable.

I.1 LINEAGE GRAPH CONSTRUCTION PROMPT

The lineage graph construction prompts are used by the paper type classifier and the dual-source lineage extractor. The paper type classification prompt determines the primary contribution type of each paper and retains methodology papers for downstream lineage extraction. The single-source lineage extraction prompt identifies experiment-grounded lineage relations from one parsed paper source by extracting prior methods that are explicitly used as empirical comparison baselines and normalizing them to full reference paper titles. When the `MinerU` and `GROBID` candidate sets conflict, the dual-source conflict resolution prompt performs candidate-restricted integration by re-checking the candidates against the `MinerU` paper content and retaining only evidence-supported baseline titles.

Paper Type Classification

System Prompt:
You are a strict research paper type classifier. Determine the PRIMARY contribution type of the paper.
Definitions:

- methodology: The paper proposes a novel model, algorithm, technical framework, or system. Experiments or datasets may be included, but the core contribution is a new technical method.
- benchmark: The paper primarily introduces a new dataset, benchmark, evaluation protocol, or leaderboard. The main contribution is evaluation infrastructure rather than a new technical method.
- survey: The paper systematically reviews, summarizes, or categorizes existing literature. It does NOT introduce a novel algorithm or technical framework.
- others: Anything that does not clearly fit the above categories.

Priority rules:

1. If the paper proposes a novel model, algorithm, or technical framework, classify it as 'methodology' even if it also includes experiments or introduces a dataset.
2. Only classify as 'benchmark' if the primary contribution is dataset or evaluation construction without a new method.
3. Only classify as 'survey' if the paper synthesizes existing work and does not propose a new method.
4. If unsure, choose 'others'.

Respond ONLY with a JSON object of the form:
{"paper_type": "methodology|benchmark|survey|others"}.

User Prompt:
Title: `[title]`
Abstract: `[abstract]`
Introduction: `[introduction]`

Single-Source Lineage Extraction

System Prompt:
You are an expert at extracting experimental baselines from academic papers. Identify only methods that are genuinely used as comparison baselines in experiments. Exclude the proposed method, its variants, ablations, datasets, metrics, tools, and methods mentioned only in background discussion. Return only a JSON object.

User Prompt:
Extraction rules:

1. Scope: Extract only methods that are explicitly compared in experimental evaluation, quantitative results, qualitative comparison, or implementation/evaluation setup tied to reported comparisons.
2. Valid baseline: A baseline must be a separate method/model/system that the paper compares against. Keep methods appearing in comparison tables, result figures, experimental setup, or evaluation text.
3. Exclude: the proposed method, ablation variants, internal modules, training tricks, datasets, metrics, loss terms, tools, and related-work citations that are not actually evaluated.
4. References are supporting evidence. The output must be the full paper title of the cited baseline work from the References section, not a method name, model nickname, abbreviation, or benchmark label.
5. If several baseline method names or variants in the paper map to the same cited paper, output that paper title only once.
6. Use references only to map evidence-supported compared methods to full paper titles. Do not output a paper title unless some provided evidence unit directly supports that method as an evaluated comparison baseline.
7. Method abbreviations, aliases, or shorthand names in evidence units may correspond to full paper titles in the references. Resolve them conservatively.
8. Do not output papers mentioned only as background, inspiration, task framing, benchmark context, or general related work.
9. Do not output generic backbone, toolkit, dataset, or optimization papers unless the evidence units explicitly present them as compared baselines.
10. Some papers are theoretical papers or do not use standard baseline comparisons. If no valid baselines are found, return an empty list.
11. Output only JSON in the format {"baseline_titles": ["Full Reference Paper Title A", "Full Reference Paper Title B"]}. Return {"baseline_titles": []} if none are found.

Paper content:
Source mode: `[source_mode]`
`[paper_content]`
Return only a JSON object with key baseline_titles.

I.2 Query Generation Prompt

The query generation prompt is used to construct multi-facet retrieval queries for LINEAGE. Given a paper title and abstract, the prompt generates short academic retrieval queries that describe the paper's research problem, task setting or application context, and method family or technical mechanism. These generated queries instantiate the topical, contextual, and methodological facets, while the title and title-context queries are directly derived from the original metadata.

Dual-Source Conflict Resolution

System Prompt:

You are an expert at extracting experimental baselines from academic papers. Identify only methods that are genuinely used as comparison baselines in experiments. Exclude the proposed method, its variants, ablations, datasets, metrics, tools, and methods mentioned only in background discussion. Return only a JSON object.

User Prompt:

Extraction rules:

`[extraction_rules]`

The baseline lists extracted from two parser sources conflict. Re-check the MinerU paper content and keep only baseline paper titles that are truly supported. Prefer a supported subset from the provided candidate titles. Do not invent new titles.

MinerU paper content:

`[mineru_paper_content]`

MinerU candidate titles:

`[mineru_candidate_titles]`

GROBID candidate titles:

`[grobid_candidate_titles]`

Return only JSON: {"baseline_titles": [str, ...]}.

I.3 LLM SEARCH PROMPT

The LLM search prompt is used to evaluate LLM-based web search baselines on LINEAGE. Given the query paper metadata and a retrieval query, the model is asked to search for real methodology papers that match the query while excluding the query paper itself. The prompt prioritizes papers that propose concrete models, algorithms, methods, or technical frameworks, while discouraging surveys, benchmarks, datasets, and purely analytical papers. It constrains returned papers to the allowed venues and the 2018–2025 publication range, and requires the model to output only a ranked JSON list of paper titles.

Query Generation

System Prompt:
You generate short academic retrieval queries from paper title and abstract. The query should be used to retrieve related papers in the same research area, not the source paper itself.
`[style_instruction]`
Generate one query for each topic below:

- abstract: Describe the paper's research problem and broad technical direction.
- background: Describe the task setting, motivation, or application context.
- methodology: Describe the method family, modeling idea, or technical mechanism.

Requirements:

- Keep each query short.
- Keep 1 to 2 concrete domain anchors, chosen from problem setting, method family, or technical mechanism.
- Use shared research terms instead of title-like, paper-specific, self-referential, or filler wording; if a method name is too specific, replace it with its broader method family or mechanism, while preserving the main technical topic.
- Background queries should contain the problem context; methodology queries should contain the method family or core technical mechanism.
- Do not add facts not present in the given title or abstract.
- If unsure, make the query slightly broader rather than narrower.
- Return valid JSON only.
- Use exactly these keys: `[expected_fields]`.

User Prompt:
Title: `[title]`
Abstract: `[abstract]`

LLM Search Prompt

System Prompt:
You are an academic paper search assistant focused on methodology papers. Use a web search to find real methodology papers matching the search query. Return only JSON.
Prioritize papers that propose a concrete model, algorithm, method, or technical framework. Avoid surveys, benchmarks, datasets, and purely analytical papers unless they introduce a clear new method. The result list must be ranked from most relevant to least relevant. Every returned paper must be published in one of the allowed venues and between 2018 and 2025 inclusive. Do not include the query paper itself. Return paper titles only; do not return venue, year, URL, reason, abstract, authors, or any other field. Do not include any extra text before or after the JSON object. Do not use the thinking mode.

User Prompt:
Query paper: `[query_paper]`
Search query: `[search_query]`
Allowed venues: `[allowed_venues]`
Return at least `[top_k]` ranked paper titles unless fewer than `[top_k]` valid matching papers exist. Return JSON with this exact shape: { "titles": ["most relevant paper title", "next most relevant paper title"] } Do not include any other text, markdown, explanation, code fences, or additional keys.