Topic Is Not Agenda: What Text Embeddings Miss in Scientific Retrieval and What Citation Structure Adds

Anonymous authors
Paper under double-blind review

Abstract

Research agents retrieve candidates by text similarity and read only the top of the list, so what text similarity misses, they never see. In this paper we measure how far text similarity is from finding papers that share a study's problem and approach, using the documents that authors cite as gold, and we propose a method that uses the citations of nearby papers to close part of this gap. We test on a public must-cite benchmark (MasterSet), a 4M-paper corpus in eight scientific domains, and US patents. The top 100 by text similarity miss 61% of the most essential papers an author cites, the baselines, benchmarks and methods its experiments build on. In the 4M-paper corpus, 30.8% of the cited papers that it holds lie beyond cosine rank 1,000, where no reranker of the text candidates can reach them, and updating the query vector does not help on average. Text embeddings find the topic of a paper, not its specific problem and approach. The method, neighbourhood citation retrieval, asks the query's nearest text neighbours what they cite, adds those documents as candidates, and ranks them by the neighbours' citations. Its training-free form, the neighbour vote, cuts these misses to 43%, and a learned form cuts them to 35%. On the same MasterSet candidates the vote beats citation popularity and a trained Citeomatic-style reranker, and adding the vote and related neighbourhood signals to that reranker raises R@100 from .556 to .660. Fine-tuning the encoder on citations does not replace the vote, which still adds to every fine-tuned encoder we tested. We also analyse where the gain comes from with several controls, including the removal of the query's own organisation (its co-authors or its assignee) from its neighbours. Because it needs no training and uses new citations as soon as they appear, the approach can serve as a retrieval interface for research agents in collections where cosine similarity is saturated, such as scientific papers and patents, where many documents share a topic and text alone cannot tell which of them share a study's problem and approach.

1 Introduction

Finding papers that are close to a study at the level of its details, the same problem or the same approach, has always been hard. Papers in one field share most of their keywords, and many papers solve the same problem in different ways or apply the same method to different problems, so keyword search (Robertson & Zaragoza, 2009) returns many papers that match the words but not the work. Dense text embeddings (Karpukhin et al., 2020; Cohan et al., 2020) match meaning instead of words and now drive most literature search. Tools such as Elicit and Consensus and research agents such as PaperQA and OpenScholar (Lála et al., 2023; Asai et al., 2026) retrieve candidates by keyword or text similarity and read the top of the list.

Text similarity, however, reaches a limit in exactly this setting. In a mature field hundreds of papers share the topic of a query and have titles and abstracts that look alike, so cosine similarity is saturated. It cannot tell which of them share the query's specific problem and approach, its agenda, and which share only its topic. We measure this gap with the documents that authors cite as gold, since an author cites the papers that the work builds on

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.

or compares against. The gap is large. The top 100 by text similarity miss 61% of the most essential papers an author cites, the baselines, benchmarks and methods its experiments build on (Ratul et al., 2026), and in a 4M-paper corpus 30.8% of the cited papers that it holds lie beyond cosine rank 1,000. Patent search meets the same limit, and in semiconductor patents cosine places only .164 of the applicant-cited prior art in the top 100. A researcher who pages further recovers part of what is missing. An agent sees only the candidates its retriever returns, so a related-work section or a novelty check built on them omits much of what the field would expect. Better text alone does not remove the gap, since neither updating the query vector nor fine-tuning the encoder on citations closes it (Sections 4 and 5).

Retrieval therefore needs a signal that does not come from the text. Citations are such a signal. Each citation is a judgment by the author of a nearby paper about what is relevant, and it separates papers that the text cannot. Most papers follow the same arc, a problem, an approach and a result. The nearest neighbours of a paper share part of this arc, the same problem in some cases and the same family of approaches in others. A paper that proposes a new approach to a problem needs the other solutions to that problem, while a paper that changes one step of an existing approach needs the work behind that approach. Corpus-wide citation counts rank the same papers first for every query and cannot tell these cases apart. The citations of the query's neighbours can, because they come from the papers that sit where the query sits. We propose neighbourhood citation retrieval, which adds the documents cited by the query's nearest text neighbours as candidates and ranks them by the neighbours' citations (Section 5). Its simplest form, the neighbour vote, counts how many neighbours cite a document. We also analyse where the gain comes from. We split the gain of fusing cosine with both popularity and the vote into a global and a local share with Shapley values, and trace the local share either to the query's own organisation, a lab or a company, or to the wider community (Section 6).

This paper makes three contributions.

- The gap. A measurement of what text similarity misses on a must-cite benchmark, a 4M-paper corpus and US patents, an analysis of four embeddings that shows where cosine similarity saturates, and tests of whether updating the query vector or fine-tuning the encoder on citations closes the gap (Sections 4 and 5).
- Neighbourhood citation retrieval. Ranking the candidates by the neighbour vote, a local citation count of how many of the query's nearest neighbours cite each one, raises MasterSet R@100 on the same encoder from .392 to .569 without training and to .650 with a learned reranker. On the same candidates the vote beats corpus-wide popularity and a trained Citeomatic-style reranker (Section 5).
- Where the gain comes from. Global and local shares under organisation controls, each tested against matched random removal. In science the local share survives the co-author control, and in patents it runs through same-assignee neighbours and, for pharmaceutical applicant citations, reverses the comparison with popularity (Section 6).

2 Related Work

Citation recommendation and scientific retrieval. Citation recommendation uses the references of a paper as gold (Strohman et al., 2007; Bhagavatula et al., 2018). Scientific embeddings learn from citations (Cohan et al., 2020; Ostendorff et al., 2022; Singh et al., 2023), and recent benchmarks evaluate literature search (Ajith et al., 2024) and must-cite recommendation (Ratul et al., 2026). These studies compare methods on this gold, and we use the same gold.

Query-dependent link analysis and graph signals. HITS builds a set of documents for each query from a text search and ranks them by the links inside that set (Kleinberg, 1999). Citation recommendation ranks candidates with corpus-wide citation counts. Strohman et al. (2007) combine them with text similarity, Katz distance and co-citation with the

neighbour set, which needs the documents that cite each candidate, and Citeomatic (Bhagavatula et al., 2018) combines them with text similarity in a neural model. Neither counts how many of the query's own neighbours cite a candidate, and this local citation count, the neighbour vote, is the ranking signal of our method. Collaborative filtering over citations (McNee et al., 2002) starts from the citations a paper already has, and adaptive re-ranking (MacAvaney et al., 2022) follows a graph of similar documents while it reranks. Co-citation (Small, 1973), bibliographic coupling (Kessler, 1963), personalised PageRank (Haveliwala, 2002) and cluster-based retrieval (van Rijsbergen, 1979; Kurland & Lee, 2004) use corpus structure, and relevance feedback (Rocchio, 1971) and pseudo-relevance feedback (Lavrenko & Croft, 2001; Yu et al., 2021) move the query toward relevant or top-ranked documents. Section 5 shows that the vote, not the candidate set alone, carries the gain.

Limits of single-vector retrieval. Weller et al. (2026) bound the top-k sets that a single vector can represent and find multi-vector models such as ColBERT (Khattab & Zaharia, 2020) less limited. Appendix E fits one or several vectors to reference sets.

Text and citation clustering. Scientometrics compares citation-based (Boyack & Klavans, 2010; Klavans & Boyack, 2017) and text-based (Boyack et al., 2011) clusterings of the literature and builds publication-level classifications from citation links (Waltman & van Eck, 2012). We use such a partition to audit embedding neighbourhoods.

Controls. Degree-preserving rewiring (Maslov & Sneppen, 2002) is a standard null model for networks. Patent studies separate applicant and examiner citations (Alcácer & Gittelman, 2006). Shapley values split a total among its parts (Shapley, 1953). Graph-based retrieval is usually evaluated by its total gain, and to our knowledge no evaluation has separated the part of this gain that comes from the query's own organisation. Section 6 separates it with organisation controls, each against a matched random control.

3 Data and Baselines

Corpora and gold. The gold is the documents that the query's authors cite, the standard gold of citation recommendation. The gold is the in-corpus references on the paper corpus, the must-cite (Type I) labels on MasterSet, and applicant or examiner citations for patents. The paper corpus has 4M papers in eight domains, built from OpenAlex (Priem et al., 2022), PubMed, Semantic Scholar and publisher pages (Appendix B). We evaluate 1,171 query papers, sampled by domain and citation quintile, with 6,361 references inside the corpus. MasterSet is a public must-cite benchmark with 6,105 queries from 2025 that have Type I labels, Type I labels that mark the baselines a paper compares against, the sources of its benchmarks and datasets, and the methods it directly builds on, and a pool of 67,761 papers from 2018–2024 (Ratul et al., 2026). We follow its protocol and call the paper corpus and MasterSet science. US patents are 388,571 patents in pharmaceuticals (A61K) and semiconductors (H01L), with 1,173 and 1,497 query patents. Their gold is their in-class citations, added by the applicant or, as a third party, by the examiner. All corpora use Gemini Embedding 001 (Lee et al., 2025) on title and abstract unless stated. The metric is R@100, the share of a query's gold in the top 100, and differences are in points (R@100 $\times$ 100). Shares and control results come with 95% bootstrap intervals over queries, and Table 6 in Appendix B lists the embedding and the time setting behind each result.

Baselines and rules. Cosine ranks the corpus by cosine similarity to the query. Popularity ranks the cosine top 1,000 by citation count and is fused with cosine by reciprocal rank fusion (Cormack et al., 2009) with $k = 60$. On the candidates of our method we also compare popularity, co-citation with the neighbours, bibliographic coupling with the neighbours' references and personalised PageRank from the neighbours, each fused with cosine in the same way. The query's own references and citers never act as its neighbours. The time constraint removes every document published after the query, or for patents after the query's filing date, from the pool, the neighbours and the vote, and counts only earlier citations. MasterSet meets it by construction.

4 Text Similarity Is Saturated

The gap is large. The top 100 by cosine miss 61% of the MasterSet gold, the baselines, benchmarks and methods that a paper's experiments build on (R@100 .392). In the 4M-paper corpus, 30.8% of the query papers' in-corpus references rank below 1,000 (26.2% when only earlier documents are ranked), and only 16.5% reach the top 10 (Figure 1). No reranker of the cosine top 1,000 can reach the references below it.

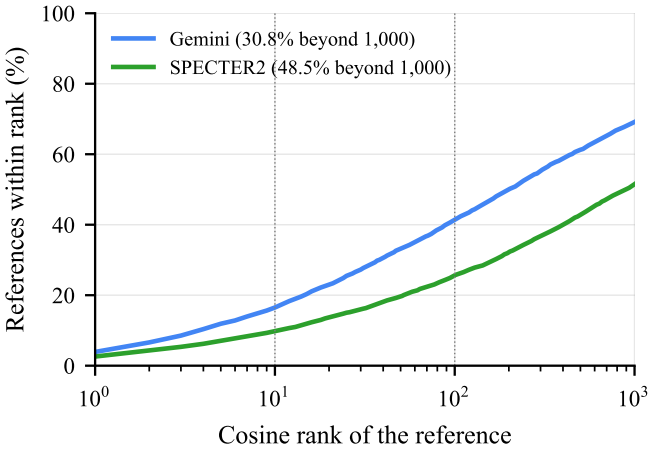


Figure 1: Share of the query papers' references within each cosine rank over the 4M corpus, for Gemini and SPECTER2, pooled over all references and without the time constraint. The curves end at rank 1,000. R@100 elsewhere is a mean over queries, which gives queries with few references more weight.

Cosine similarity is saturated. Inside the cosine top 1,000, the full embeddings add little to cosine in separating the cited papers from the similar papers around them. A gradient-boosted classifier that sees the query and candidate embeddings (their cosine, element-wise product and absolute difference), trained with cross-validation over queries, separates them with an AUC of .738 against .718 for cosine and rank alone. The nearest neighbours of a paper share its sub-field but mostly not its finer community. Four embeddings place 45–52% of a paper's ten nearest neighbours in its citation sub-field but only 15–21% in its finer citation community, on average over domains (Figure 2). A finer community has fewer members, so we also compare each rate with the most that a paper's ten neighbours could reach. Relative to this ceiling the embeddings reach 46–54% at L1 but only 22–30% at L2, and the drop holds in every model, domain and L2 resolution (Appendices A and D). The communities come from a graph over the 3.58M papers of the corpus with direct citation, bibliographic coupling and co-citation edges, partitioned with Leiden and the Constant Potts Model (Traag et al., 2019). L1 communities are sub-fields ($\gamma = 10^{-4}$), and a second pass inside each L1 gives L2 communities ($\gamma = 10^{-2}$, mean non-singleton size 19). The audit uses four embeddings (Gemini, Qwen3-Embedding-8B and 0.6B (Zhang et al., 2025), SPECTER2 (Singh et al., 2023)), and every paper in a pool of 200K papers per domain acts as a query.

Better text does not close the gap. A query built from half of a paper's references and scored on the other half does not beat the paper's own vector on average, and it helps only when five or more references are known (Tables 14 and 15, Appendix E). Moving the query vector toward its top neighbours by pseudo-relevance feedback does not help either (.357 against .365 with bge-base on queries with at least three references and no time constraint, Appendix E). Fine-tuning bge-base (Xiao et al., 2024) on citations from papers up to 2018 raises cosine R@100 on the 765 later queries only from .420 to at most .471, so more than half of the gold still stays outside the top 100.

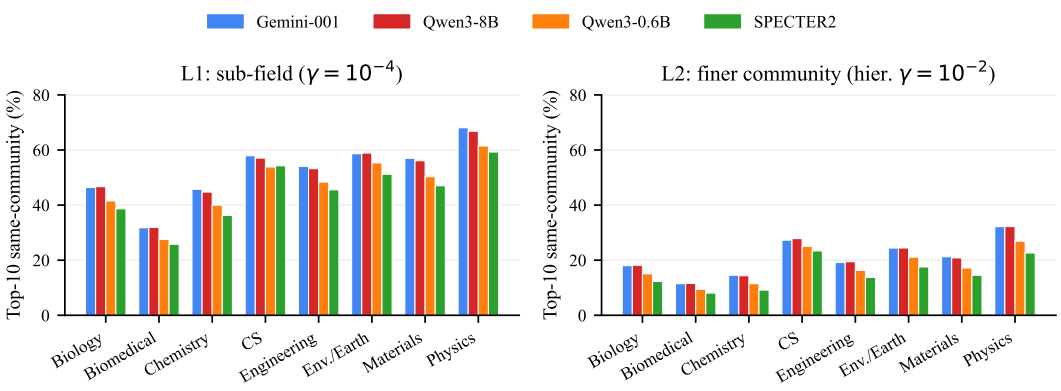


Figure 2: Top-10 embedding neighbours in the query's citation community, by domain and model.

5 Neighbourhood Citation Retrieval

Neighbourhood citation retrieval asks the query's text neighbours what they cite (Figure 3). It takes the query's 100 nearest neighbours and adds every document they cite to the cosine top 1,000 as a candidate. Its simplest form, the neighbour vote, gives a document one vote for each neighbour that cites it and fuses the vote ranking with the cosine ranking by reciprocal rank fusion ($k = 60$), where an added candidate, which has no rank in the cosine top 1,000, enters through its vote rank alone. Adding candidates alone does not change the ranking, because in a cosine ranking the added documents fall below the pool, and the neighbours' citations are what lift them. The vote is a local citation count. It depends on what the papers near the query cite, not on how often a document is cited in the whole corpus. Neighbours that share the query's problem vote for the other solutions to it, and neighbours that share its approach vote for the work behind that approach, so the vote follows whichever part of the query's problem and approach its neighbourhood shares. It needs no training, uses new citations as soon as they enter the graph, and can sit on any text first stage. A learned form ranks the same candidates from cosine, popularity and the neighbourhood signals (vote, co-citation, coupling and personalised PageRank from the neighbours) with a gradient-boosted classifier, trained with five-fold cross-validation over queries (Appendix E).

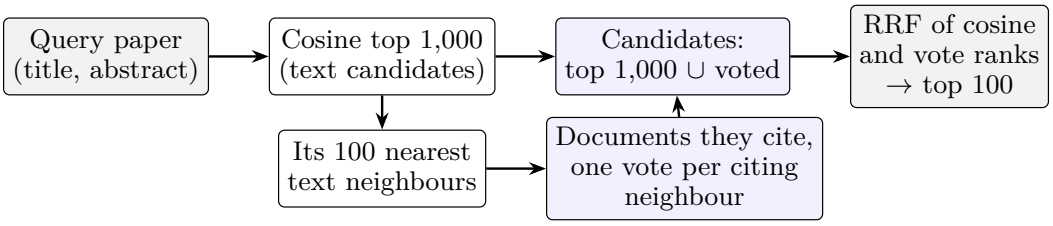


Figure 3: The neighbour vote. The query's nearest text neighbours vote for the documents they cite, the voted documents join the text candidates, and the vote and cosine rankings are fused.

It recovers what text misses. Cosine with Gemini already exceeds the best published MasterSet system (.392 against .383, Ratul et al., 2026), so we compare methods on the same encoder. The vote raises MasterSet R@100 from .392 to .569, and the learned form to .650, which cuts the missed gold to 43% and 35% (Table 1). On the paper corpus the vote raises R@100 from .558 to .623, and the learned form to .693. With co-author neighbours removed, the vote brings 34.7% of the MasterSet gold beyond cosine rank 1,000 into the

top 100, against 24.6% when the 200 most-cited documents of the pool are added as candidates and ranked by popularity. An agent that reads the top 100 by cosine alone would not see these documents, and on the paper corpus the least-cited recovered references of computer-science queries include methods, tools and datasets that the query uses (Table 17, Appendix F).

Table 1: R@100 of cosine, cosine fused with popularity over the cosine top 1,000, the neighbour vote, and the learned form (five-fold cross-validation over queries). MasterSet uses its Type I labels, on which the best published system reaches .383. The paper corpus is time-constrained.

Corpus	Cosine	+ popularity (top 1,000)	Neighbour vote	Learned form
MasterSet	.392	.477	.569	.650
Papers (time-constrained)	.558	.588	.623	.693

The vote beats popularity. On the same candidates in science, the vote and co-citation with the neighbours beat popularity. On MasterSet the vote reaches .569 and co-citation .565, against .489 for popularity, and on the paper corpus the vote reaches .623 against .600 (Table 2). In patents popularity can match or beat them (Section 6). The vote is the simplest of these signals, one hop from the neighbours' reference lists, and it is as good as co-citation, which needs the documents that cite the neighbours. The signal lies in the citation edges themselves. When the edges are shuffled with every node's in- and out-degree kept, MasterSet R@100 falls from .569 to .412, close to cosine (.392). The learned form gains through the same signals. Trained on text features only, cosine, cosine rank and pool membership, it stays near cosine (.392 and .553). Trained on text and popularity, it reaches .532 on MasterSet and .613 on the paper corpus, and trained on text and the neighbourhood signals without popularity, it reaches .648 and .694, as high as with all features. On the same candidates and with the same learner, the neighbourhood signals beat popularity by +11.6 points [+11.1, +12.1] and +8.1 points [+7.0, +9.2].

Table 2: R@100 of five methods, each fused with cosine, on the same candidates: the cosine pool plus the neighbours' citations (time-constrained for papers and patents). Popularity thus differs from Tables 1 and 4.

Condition	Control	Cosine	Vote	Pop. (all cand.)	Co-cit.	PPR	Bib.
MasterSet	none	.392	.569	.489	.565	.399	.358
	+ co-author	.392	.565	.489	.563	.402	.356
Papers	none	.558	.623	.600	.612	.534	.532
	+ co-author	.558	.614	.598	.608	.526	.526
Semicond., applicant	family	.164	.284	.300	.232	.182	.167
	+ assignee	.164	.174	.206	.181	.136	.158
Pharma, applicant	family	.478	.544	.542	.516	.474	.470
	+ assignee	.478	.501	.529	.496	.452	.460
Semicond., examiner	family	.161	.177	.173	.179	.143	.163
	+ assignee	.161	.166	.171	.172	.131	.155
Pharma, examiner	family	.445	.443	.426	.442	.424	.448
	+ assignee	.445	.436	.425	.436	.420	.438

It beats a Citeomatic-style reranker. Citeomatic (Bhagavatula et al., 2018) ranks candidates from the same source, the citations of the query's text neighbours, with a trained model over field-wise text similarity, word overlap, authors, venue and corpus-wide citation counts. We train this model with the same learner on the same MasterSet candidates, with TF-IDF vectors of the title and abstract in place of learned field embeddings (Appendix E), and it reaches .556. Here the vote reaches .572 (.569 in Table 1), since here the added candidates also get a cosine term, while Table 1 fuses them by vote rank alone. The training-free

vote beats the reranker by +1.6 points [+1.1, +2.1], and the neighbourhood signals in the same learner beat it by +9.2 points [+8.6, +9.7]. Added to the Citeomatic-style features, the neighbourhood signals raise R@100 by +10.4 points [+9.9, +10.9] to .660, the best result on MasterSet.

The signal works as candidates, not as a better query. Moving the query vector toward the neighbours' top-voted documents does not help (.358 against .365, Appendix E), so the neighbours' citations help when they add candidates and rank them, not when they rewrite the query.

Fine-tuning does not replace it. The encoders fine-tuned on citations up to 2018 in Section 4 move cosine 12–53% of the way to the .516 that the vote gives the base model (Table 3). Training on the neighbours' own top-voted documents moves it least (12%, 95% CI −4% to 28%), and on each of the four encoders the vote still adds 6.3–7.0 points.

Table 3: Fine-tuning bge-base on citations from papers up to 2018, with R@100 on the 765 queries from 2019 on (time-constrained). Absorbed is the share of the base model's vote gain that fine-tuning gives to cosine. Table 16 in Appendix E adds confidence intervals and more encoders.

Training data	Cosine	+ vote	Absorbed
None (bge-base)	.420	.516	–
References	.458	.528	39%
References, hard negatives	.471	.534	53%
References and votes	.452	.517	33%
Votes only	.432	.501	12%

It adds on a keyword first stage. On BM25 (Robertson & Zaragoza, 2009) candidates reranked by embedding, the vote raises R@100 from .356 to .465, above dense retrieval with the same short query over the whole corpus (.408, Appendix G).

6 Where the Gain Comes From

The vote's gain can also come from the query's own organisation, a lab or a company whose papers cite each other. We split the gain into a share that popularity also gives and a local share, and test whether the local share survives the removal of the query's organisation.

Table 4: R@100 of cosine (C), cosine with popularity (P), cosine with the vote (V) and the three-way fusion (PV), with the Shapley shares in points, before the co-author or assignee control. Popularity ranks the cosine top 1,000. Patent rows are time-constrained and use the family control.

Corpus	C	P	V	PV	Global share	Local share
MasterSet	.392	.477	.569	.520	+1.8 [+1.4, +2.2]	+11.0 [+10.6, +11.4]
Papers (time-constrained)	.558	.588	.623	.613	+1.0 [+0.2, +1.8]	+4.5 [+3.7, +5.3]
Semicond., applicant	.164	.207	.284	.237	−0.2 [−0.9, +0.4]	+7.5 [+6.6, +8.4]
Pharma, applicant	.478	.533	.544	.545	+2.8 [+2.1, +3.6]	+4.0 [+3.4, +4.7]
Semicond., examiner	.161	.170	.177	.178	+0.5 [0.0, +1.0]	+1.2 [+0.8, +1.6]
Pharma, examiner	.445	.426	.443	.439	−1.1 [−1.9, −0.3]	+0.6 [+0.1, +1.0]

Shares and controls. Let C, P, V and PV be the R@100 of cosine, cosine with popularity, cosine with the vote, and the three-way fusion. The Shapley values (Shapley, 1953) of the two additions are global $= \frac{1}{2}[(PV-V)+(P-C)]$ and local $= \frac{1}{2}[(PV-P)+(V-C)]$, and they sum to $PV-C$ (Table 4). Popularity uses corpus-wide citation counts, hence global, and the vote uses only the citations of the query's 100 nearest neighbours, hence local. The three-way fusion serves only this split, and because popularity cannot add candidates, the local share

credits the vote with the candidates it adds (Table 8 in Appendix C lets popularity rank the same candidates). The co-author control removes neighbours written by anyone within two co-author hops of the query's authors, which is meant to cover their advisors, students, lab members and the labs' collaborators (Appendix B). For patents, the family control removes every document that shares the query's family or one of its priority applications, and the assignee control also removes neighbours that share the query's assignee. We call the neighbours that these controls remove the organisation unit, and refill removed neighbours with the next cosine neighbours so that every query keeps 100. Each organisation control is compared with a matched random control, which removes the same number of neighbours at random, and a rank-matched control, which removes for each neighbour in the unit the other neighbour nearest to it in rank.

Science: the local share survives the co-author control. Removing the neighbours within two co-author hops of the query's authors lowers the local share from +11.0 to +10.7 points on MasterSet and from +4.5 to +3.7 on the paper corpus, close to removing the same number of random neighbours (+10.9 and +4.2, Figure 4). The co-author unit does carry a small part. The paired difference to random removal is −0.3 points [−0.4, −0.1] on MasterSet and −0.5 [−0.8, −0.2] on the paper corpus, under one point. The test can see an organisation effect, since the co-author unit is 11% of the neighbours on MasterSet and 9% on the paper corpus, about twice the share of same-assignee neighbours in pharmaceutical patents (5%), whose removal takes most of the local share there. In science the local gain thus comes mostly from authors outside the query's own circle (Table 10 in Appendix C splits it by self-citations).

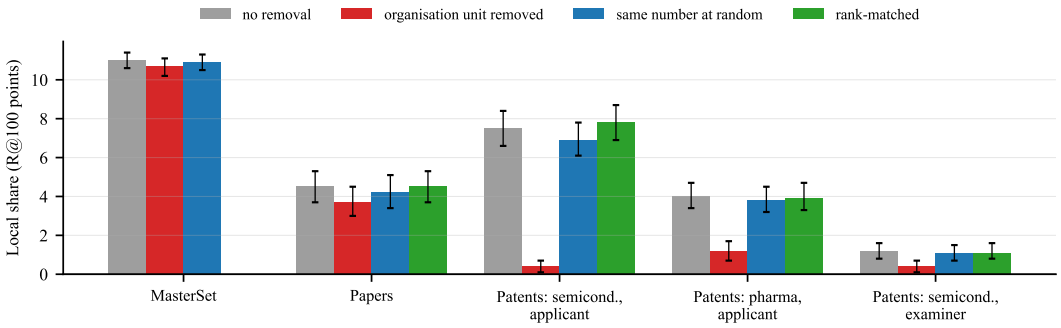


Figure 4: Local share of the citation-structure gain (R@100 points, 95% bootstrap CI) without removal, with the organisation unit removed (co-author 2-hop for MasterSet and papers, assignee for patents), and with the same number of random or rank-matched neighbours removed (rank-matched not run on MasterSet). Patents are time-constrained and use the family control. Table 7 (Appendix C) gives the values and adds pharmaceutical examiner citations.

Patents: what the method recovers depends on whose citations define relevance. With applicant citations as gold, the local share runs through the applicant's own portfolio. In semiconductors it falls from +7.5 to +0.4 points when same-assignee neighbours are removed (+6.9 under random removal), and it falls from +9.1 to +0.8 even for prior art of other assignees (Table 10, Appendix C). This drop holds across RRF k and with popularity on the vote's candidates (Table 8, Appendix C). With examiner citations, a third party's choice, the local share under the time constraint is small (+1.2 and +0.6 points). The assignee control also moves the same-candidate comparison toward popularity in all four patent conditions, and for pharmaceutical applicant citations the vote's lead of +0.2 points turns into −2.8 (Table 2).

7 Discussion

The candidates decide what an agent can find. An agent does not look past what its retriever returns, and many cited papers lie beyond the reach of any reranker of the text candidates (Section 4). Neighbourhood citation retrieval adds the documents that the neighbours cite to the candidates, at query time and with the newest citations. Later steps, such as iterative query refinement or reranking, can then order the top of the list.

For evaluation, the total gain is not enough. In patents, two choices change both the gain and the comparison between methods, namely whose citations define relevance and whether the query's own organisation may help (Section 6). An evaluation of such retrieval should therefore report the global and local shares and test organisation controls against matched random controls. The same protocol applies wherever documents carry links that people made, such as case law, code dependencies and web links.

The results cover title and abstract text, eight scientific domains, one must-cite benchmark and two patent classes.

8 Conclusion

Text similarity finds the topic of a paper but not its specific problem and approach. In a mature field many papers share a topic and look alike, so cosine similarity is saturated, and the top 100 by text similarity miss 61% of the most essential papers an author cites. Neither updating the query vector nor fine-tuning the encoder on citations closes this gap.

The signal that text lacks is local. Neighbourhood citation retrieval asks the query's nearest text neighbours what they cite and ranks the candidates by their citations. Its training-free form, the neighbour vote, cuts the missed essential papers to 43%, and a learned form cuts them to 35%. On the same candidates in science the vote beats corpus-wide popularity and a trained Citeomatic-style reranker, and it still adds on encoders fine-tuned on citations. What the papers near a query cite separates the papers that its text cannot.

Because the vote needs no training and uses new citations as soon as they appear, it can serve as a retrieval interface for research agents in collections where cosine similarity is saturated, such as scientific literature and patents. Controls that remove the query's own co-authors or assignee from its neighbours show that in science the gain comes mostly from the wider community, while in patents with applicant citations as gold it runs through the applicant's own portfolio. Where text alone cannot tell papers apart, the judgments that nearby authors have already made through their citations can.

AI use statement

In this work, we used generative AI tools to implement methods, as AI coding assistants wrote most of the experiment, analysis and plotting code under our direction, to give feedback on and propose parts of the experimental design, and to help interpret results. An LLM is also part of two pipelines, where it extracted named entities from query abstracts (Appendix E) and generated a synthetic set of short keyword queries, one per query paper (Appendix G). We have not used generative AI tools to develop theoretical models, formulate or prove mathematical claims, or translate, and qualitative or thematic data analysis is not applicable to this work. Additionally, generative AI tools helped us draft the paper and reviewed drafts as mock reviewers. We have reviewed all AI-assisted work. We re-ran every result from the stored outputs, every number in the text and tables comes from one file of macros that a script builds from the result files and the logs of the audit runs, and we checked each reference against its original source. We take responsibility for the final content of this work, including text, claims and artifacts produced with the aid of generative AI.

Ethics statement

The data are public bibliographic metadata and public patent records. Author identities are used only for the co-author control, the self-citation split and the shared-author test. We do not release author identities or co-author graphs. The work involves no human subjects.

Reproducibility statement

Section 3 defines the corpora, gold and baselines, Section 5 the method, and Section 6 the shares and controls. Appendix B describes the data sources, Appendix E the fitted-vector, held-out, fine-tuning and learned-form protocols, and Appendix G the keyword front end. The supplementary material contains the code for every experiment, the OpenAlex identifiers of the corpus, the query and gold lists, the per-query results, and the scripts that generate every number, table and figure of the paper from those results.

References

Anirudh Ajith et al. LitSearch: A retrieval benchmark for scientific literature search. In Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 15068–15083, 2024.

Juan Alcácer and Michelle Gittelman. Patent citations as a measure of knowledge flows: The influence of examiner citations. The Review of Economics and Statistics, 88(4):774–779, 2006.

Akari Asai et al. Synthesizing scientific literature with retrieval-augmented language models. Nature, 650(8103):857–863, 2026.

Chandra Bhagavatula et al. Content-based citation recommendation. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), Volume 1 (Long Papers), pp. 238–251, 2018.

Vincent D. Blondel et al. Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment, 2008(10):P10008, 2008.

Kevin W. Boyack and Richard Klavans. Co-citation analysis, bibliographic coupling, and direct citation: Which citation approach represents the research front most accurately? Journal of the American Society for Information Science and Technology, 61(12):2389–2404, 2010.

Kevin W. Boyack et al. Clustering more than two million biomedical publications: Comparing the accuracies of nine text-based similarity approaches. PLoS ONE, 6(3):e18029, 2011.

Arman Cohan et al. SPECTER: Document-level representation learning using citation-informed transformers. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL), pp. 2270–2282, 2020.

Gordon V. Cormack et al. Reciprocal rank fusion outperforms Condorcet and individual rank learning methods. In Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), pp. 758–759, 2009.

Taher H. Haveliwala. Topic-sensitive PageRank. In Proceedings of the 11th International Conference on World Wide Web (WWW), pp. 517–526, 2002.

Vladimir Karpukhin et al. Dense passage retrieval for open-domain question answering. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 6769–6781, 2020.

M. M. Kessler. Bibliographic coupling between scientific papers. American Documentation, 14(1):10–25, 1963.

Omar Khattab and Matei Zaharia. ColBERT: Efficient and effective passage search via contextualized late interaction over BERT. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), pp. 39–48, 2020.

Richard Klavans and Kevin W. Boyack. Which type of citation analysis generates the most accurate taxonomy of scientific and technical knowledge? Journal of the Association for Information Science and Technology, 68(4):984–998, 2017.

Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. Journal of the ACM, 46(5):604–632, 1999.

Oren Kurland and Lillian Lee. Corpus structure, language models, and ad hoc information retrieval. In Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), pp. 194–201, 2004.

Jakub Lála et al. PaperQA: Retrieval-augmented generative agent for scientific research. arXiv preprint arXiv:2312.07559, 2023.

Victor Lavrenko and W. Bruce Croft. Relevance-based language models. In Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), pp. 120–127, 2001.

Jinhyuk Lee et al. Gemini Embedding: Generalizable embeddings from Gemini. arXiv preprint arXiv:2503.07891, 2025.

Sean MacAvaney et al. Adaptive re-ranking with a corpus graph. In Proceedings of the 31st ACM International Conference on Information & Knowledge Management (CIKM), pp. 1491–1500, 2022.

Sergei Maslov and Kim Sneppen. Specificity and stability in topology of protein networks. Science, 296(5569):910–913, 2002.

Sean M. McNee et al. On the recommending of citations for research papers. In Proceedings of the 2002 ACM Conference on Computer Supported Cooperative Work (CSCW), pp. 116–125, 2002.

Malte Ostendorff et al. Neighborhood contrastive learning for scientific document representations with citation embeddings. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 11670–11688, 2022.

Jason Priem et al. OpenAlex: A fully-open index of scholarly works, authors, venues, institutions, and concepts. In Proceedings of the 26th International Conference on Science, Technology and Innovation Indicators (STI), 2022.

Md Toyaha Rahman Ratul et al. MasterSet: A large-scale benchmark for must-cite citation recommendation in the AI/ML literature. In Proceedings of the 2026 SIAM International Conference on Data Mining (SDM), 2026.

Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: BM25 and beyond. Foundations and Trends in Information Retrieval, 3(4):333–389, 2009.

J. J. Rocchio. Relevance feedback in information retrieval. In Gerard Salton (ed.), The SMART Retrieval System: Experiments in Automatic Document Processing, pp. 313–323. Prentice-Hall, 1971.

Lloyd S. Shapley. A value for n-person games. In Harold W. Kuhn and Albert W. Tucker (eds.), Contributions to the Theory of Games II, number 28 in Annals of Mathematics Studies, pp. 307–318. Princeton University Press, 1953.

Amanpreet Singh et al. SciRepEval: A multi-format benchmark for scientific document representations. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 5548–5566, 2023.

Henry Small. Co-citation in the scientific literature: A new measure of the relationship between two documents. Journal of the American Society for Information Science, 24(4): 265–269, 1973.

Trevor Strohman et al. Recommending citations for academic papers. In Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), pp. 705–706, 2007.

Vincent A. Traag et al. From Louvain to Leiden: Guaranteeing well-connected communities. Scientific Reports, 9(1):5233, 2019.

C. J. van Rijsbergen. Information Retrieval. Butterworths, 2nd edition, 1979.

Ludo Waltman and Nees Jan van Eck. A new methodology for constructing a publication-level classification system of science. Journal of the American Society for Information Science and Technology, 63(12):2378–2392, 2012.

Orion Weller et al. On the theoretical limitations of embedding-based retrieval. In Proceedings of the 14th International Conference on Learning Representations (ICLR), 2026.

Shitao Xiao et al. C-Pack: Packed resources for general Chinese embeddings. In Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), pp. 641–649, 2024.

HongChien Yu et al. Improving query representations for dense retrieval with pseudo relevance feedback. In Proceedings of the 30th ACM International Conference on Information & Knowledge Management (CIKM), pp. 3592–3596, 2021.

Yanzhao Zhang et al. Qwen3 Embedding: Advancing text embedding and reranking through foundation models. arXiv preprint arXiv:2506.05176, 2025.

A Per-domain audit

Table 5 gives the values behind Figure 2.

Table 5: Top-10 same-community rate by domain (%). Baseline is the chance that two random papers in the domain pool share a community.

Domain	Baseline (%)	Gemini	Qwen3-8B	Qwen3-0.6B	SPECTER2
L1 (sub-field, $\gamma = 10^{-4}$)					
Biology	0.456	46.4	46.7	41.5	38.7
Biomedical	0.161	31.8	31.9	27.5	25.8
Chemistry	0.566	45.7	44.7	40.0	36.2
CS	1.20	57.9	57.1	53.8	54.3
Engineering	0.360	54.0	53.2	48.4	45.5
Env./Earth	1.32	58.6	58.9	55.3	51.2
Materials	0.665	57.0	56.1	50.3	47.0
Physics	2.63	68.1	66.8	61.4	59.3
Mean		52.4	51.9	47.3	44.7
L2 (hierarchical $\gamma = 10^{-2}$)					
Biology	0.0115	18.0	18.2	15.1	12.3
Biomedical	0.0088	11.5	11.6	9.4	8.1
Chemistry	0.0111	14.6	14.4	11.5	9.1
CS	0.0279	27.3	27.8	25.0	23.4
Engineering	0.0096	19.2	19.4	16.3	13.8
Env./Earth	0.0236	24.4	24.4	21.1	17.5
Materials	0.0179	21.3	20.8	17.2	14.5
Physics	0.0462	32.2	32.2	26.9	22.6
Mean		21.0	21.1	17.8	15.2

B Data

Paper corpus. The corpus combines OpenAlex, PubMed and Semantic Scholar records, with abstracts crawled from publisher pages when all three sources lack one. Records are deduplicated by DOI and by a hash of title, first author and year. We keep original research articles from 2010 onward in SCIE-indexed journals and CORE A*/A conferences, in eight domains: biology, biomedicine, chemistry, computer science, engineering, environmental and earth sciences, materials science and physics. We sample 500K papers per domain, stratified by publication year, which gives the 4M corpus. For the citation-community graph we remove duplicates, boilerplate abstracts and papers with no edge, which leaves 3.58M papers. Citation edges inside the corpus come from the reference lists of the sources. Every paper is identified by its OpenAlex work ID, so the corpus can be rebuilt from public records. A query paper has 51 references on average in OpenAlex, and 11% of them are in the corpus; the gold is this in-corpus part.

Embeddings. SPECTER2 is allenai/specter2_base without an adapter, with title and abstract joined by its separator token, 512 tokens and [CLS] pooling.

Authors. The author disambiguation combines several matching models. Papers with more than 50 authors and authors with more than 500 papers are not expanded.

MasterSet. We use the released MasterSet data and its Type I labels (Ratul et al., 2026). The pool, the queries and the metrics follow its protocol. The vote and popularity use the citation graph among pool papers. The co-author control uses author names, since the data have no author identifiers.

Patents. We take US patents in CPC classes A61K and H01L from the public Google Patents data. The gold of a query is its in-class citations, split into applicant and examiner citations. Patent families and harmonised assignee names come from the same source.

Licenses. OpenAlex is CC0, PubMed is public domain, and Semantic Scholar data follow its API terms. Abstracts crawled from publisher pages are not redistributed. Google Patents public data follow their terms of use. We release the OpenAlex identifiers of the corpus, the query and gold lists, the patent publication numbers, the per-query results and the code under CC-BY 4.0 (data) and MIT (code).

Table 6: Embedding and time setting behind each result.

Result	Embedding	Time constraint
Audit, §4	four models	–
References outside the top 1,000, §4	Gemini, SPECTER2	both
Text separability (AUC), §4	Gemini	no
Fitted vector, held-out, §4	bge, bge fine-tuned, Gemini	no
Fine-tuning, §4–5	bge-base, bge-large, fine-tuned	both (Table 16)
Other text routes	entity match; Gemini	no
Query-vector updates	bge-base	no
Vote, methods, learned form, papers, §5	Gemini	yes
Vote, methods, learned form, Citeomatic-style, MasterSet	Gemini	by construction
Rewiring, MasterSet	Gemini	by construction
Keyword front end	BM25, Gemini	no
Shares and controls, papers, §6	Gemini	yes
Shares and controls, MasterSet	Gemini	by construction
Shares and controls, patents	Gemini	yes (filing date)

C Robustness of the shares

Table 7 gives the values of Figure 4 and adds pharmaceutical examiner citations, Table 8 varies the fusion, Table 9 drops the time constraint for patents, Table 10 splits the local share by kind of gold, and Table 11 by the number of references.

Table 7: Local share of the citation-structure gain (Shapley, R@100 points, 95% bootstrap CI), with popularity inside the cosine top 1,000. We remove the neighbours in the organisation unit, the same number of random neighbours, or the other neighbours nearest in rank to the unit's neighbours (not run on MasterSet), and refill by rank.

Corpus	No removal	Unit removed	Random	Rank-matched
Science; unit = co-author 2-hop				
MasterSet	+11.0 [+10.6, +11.4]	+10.7 [+10.2, +11.1]	+10.9 [+10.5, +11.3]	–
Papers (time-constrained)	+4.5 [+3.7, +5.3]	+3.7 [+3.0, +4.5]	+4.2 [+3.4, +5.1]	+4.5 [+3.7, +5.3]
US patents (time-constrained), family already removed; unit = assignee				
Semicond., applicant	+7.5 [+6.6, +8.4]	+0.4 [+0.1, +0.7]	+6.9 [+6.1, +7.8]	+7.8 [+6.9, +8.7]
Pharma, applicant	+4.0 [+3.4, +4.7]	+1.2 [+0.7, +1.7]	+3.8 [+3.2, +4.5]	+3.9 [+3.3, +4.7]
Semicond., examiner	+1.2 [+0.8, +1.6]	+0.4 [+0.1, +0.7]	+1.1 [+0.7, +1.5]	+1.1 [+0.8, +1.6]
Pharma, examiner	+0.6 [+0.1, +1.0]	+0.1 [−0.4, +0.6]	+0.4 [−0.1, +0.9]	+0.4 [−0.1, +0.9]

Table 8: Local share (R@100 points) at RRF $k = 60$, when popularity ranks the same candidates as the vote, and with RRF $k = 10$ or $k = 200$. The unit is co-author 2-hop for papers and assignee for patents; patent rows are time-constrained and use the family control. "Kept" is the local share after removal as a percentage of the share without removal, computed before rounding; it is not given for pharmaceutical examiner citations, whose local share stays below one point in every variant.

Corpus	Variant	No removal	Unit removed	Random	Unit kept (%)	Random kept (%)
Papers (time-constrained)	RRF $k = 60$ (main)	+4.5	+3.7	+4.2	83	94
	fixed candidates	+4.4	+3.5	+4.2	80	95
	RRF $k = 10$	+5.2	+4.3	+5.0	82	96
	RRF $k = 200$	+2.2	+1.5	+2.0	68	88
Semicond., applicant	RRF $k = 60$ (main)	+7.5	+0.4	+6.9	6	92
	fixed candidates	+6.6	+0.1	+6.0	2	92
	RRF $k = 10$	+11.1	+0.3	+10.2	3	92
	RRF $k = 200$	+1.3	−0.3	+1.1	−21	82
Pharma, applicant	RRF $k = 60$ (main)	+4.0	+1.2	+3.8	30	95
	fixed candidates	+4.6	+1.2	+4.4	25	97
	RRF $k = 10$	+4.9	+0.9	+4.7	18	95
	RRF $k = 200$	+2.8	+1.0	+2.7	35	96
Semicond., examiner	RRF $k = 60$ (main)	+1.2	+0.4	+1.1	33	90
	fixed candidates	+1.3	+0.2	+1.0	19	83
	RRF $k = 10$	+1.2	+0.1	+1.1	9	98
	RRF $k = 200$	+0.7	+0.2	+0.6	29	84
Pharma, examiner	RRF $k = 60$ (main)	+0.6	+0.1	+0.4	–	–
	fixed candidates	+0.3	−0.2	+0.1	–	–
	RRF $k = 10$	0.0	−0.5	−0.2	–	–
	RRF $k = 200$	+0.7	+0.3	+0.4	–	–

Table 9: Patent local share (R@100 points, 95% bootstrap CI) without the time constraint and with the family control only, without priority applications.

Condition	Family only	+ assignee	+ random	+ rank-matched
Semicond., applicant	+19.9 [+18.6, +21.3]	+0.9 [+0.5, +1.4]	+18.7 [+17.3, +20.1]	+20.2 [+18.8, +21.6]
Pharma, applicant	+11.9 [+11.0, +12.9]	+1.1 [+0.3, +1.8]	+11.6 [+10.6, +12.5]	+11.7 [+10.7, +12.7]
Semicond., examiner	+5.7 [+5.0, +6.4]	+1.2 [+0.7, +1.6]	+4.7 [+4.1, +5.4]	+5.5 [+4.8, +6.2]
Pharma, examiner	+3.4 [+2.6, +4.2]	−1.7 [−2.4, −0.9]	+3.0 [+2.3, +3.8]	+3.3 [+2.5, +4.1]

On MasterSet the pattern is the same for its other two label types, core relevance and mention frequency: the local share is +9.8 and +7.1 points, +9.4 and +6.7 under the co-author control, and +9.6 and +7.0 under the random control.

Table 10: Local share by kind of gold: citations that share an author (papers) or an assignee (patents, "same") with the query, and the other citations. Columns as in Table 7; "Share" is the share of the gold, and each cell averages over the queries that have that kind of gold, so both parts can exceed the local share over all queries.

Gold	Share	No removal	Unit removed	Random	Rank-matched
Papers (time-constrained); unit = co-author 2-hop					
Self-citations	16.5%	+1.8 [+0.2, +3.3]	−1.2 [−2.7, +0.2]	+0.4 [−1.0, +1.8]	+1.9 [+0.3, +3.4]
Other		+4.8 [+3.9, +5.7]	+4.4 [+3.5, +5.4]	+4.9 [+4.0, +5.8]	+4.9 [+4.0, +5.8]
US patents (time-constrained), family already removed; unit = assignee					
Semicond., applicant, same	44%	+8.0 [+6.7, +9.3]	−0.4 [−0.8, +0.1]	+7.3 [+6.0, +8.5]	+8.5 [+7.2, +9.8]
Semicond., applicant, other		+9.1 [+8.0, +10.2]	+0.8 [+0.4, +1.3]	+8.7 [+7.7, +9.8]	+9.5 [+8.4, +10.6]
Pharma, applicant, same	45%	+2.6 [+1.9, +3.4]	−0.1 [−0.7, +0.5]	+2.4 [+1.7, +3.2]	+2.6 [+1.9, +3.4]
Pharma, applicant, other		+4.8 [+3.9, +5.7]	+2.0 [+1.2, +2.7]	+4.7 [+3.8, +5.6]	+4.8 [+3.8, +5.7]
Semicond., examiner, same	39%	+1.8 [+1.1, +2.6]	−0.1 [−0.6, +0.5]	+1.0 [+0.4, +1.8]	+1.8 [+1.0, +2.5]
Semicond., examiner, other		+1.2 [+0.7, +1.8]	+0.6 [+0.2, +1.0]	+1.3 [+0.8, +2.0]	+1.1 [+0.6, +1.7]
Pharma, examiner, same	75%	−0.2 [−0.7, +0.2]	−0.7 [−1.2, −0.2]	−0.4 [−0.8, +0.1]	−0.3 [−0.8, +0.2]
Pharma, examiner, other		+3.0 [+1.5, +4.7]	+2.6 [+1.2, +4.2]	+2.9 [+1.4, +4.6]	+2.7 [+1.2, +4.3]

Table 11: Local share on the paper corpus under the time constraint, by the number of the query's references in the corpus (R@100 points, 95% bootstrap CI).

References	Queries	No removal	Co-author 2-hop removed
1–5	757	+3.9 [+2.7, +5.1]	+3.5 [+2.3, +4.6]
6–10	265	+5.1 [+3.9, +6.5]	+3.8 [+2.6, +5.2]
11–20	128	+6.1 [+4.4, +7.8]	+4.8 [+3.2, +6.5]
21+	21	+7.6 [+5.1, +10.1]	+5.1 [+2.3, +8.0]

D Partition robustness

Table 12 repeats the audit at four L2 resolutions in a separate run. Its values at $\gamma_{L2} = 10^{-2}$ are close to those in Table 5. In L1 communities sampled for this comparison, the L2 rate is 24.6% with Louvain (Blondel et al., 2008) and 17.1% with CPM; the sample differs from the pools of Table 5.

Table 12: Top-10 same-community rate (%) at L1 and at four L2 resolutions, from a separate run. The main text uses $\gamma_{L2} = 10^{-2}$.

Model	L1 (10^{-4})	L2 (5×10^{-3})	L2 (10^{-2})	L2 (2×10^{-2})	L2 (5×10^{-2})
Gemini	52.1	25.8	21.1	16.6	11.0
Qwen3-8B	51.7	25.9	21.2	16.5	10.9
Qwen3-0.6B	47.1	22.0	17.8	13.8	9.1
SPECTER2	44.4	18.9	15.1	11.7	7.6

Authorship. In 5,918 pairs of papers matched on L1, domain and the gap in publication year, pairs in the same L2 share an author in 7.05% of cases, pairs from a random part of the L1 with the size of the L2 in 0.74%, and pairs from different L2 communities in 0.27%. Among pairs with no edge of any kind in the graph, the same-L2 and random-part rates are 2.49% and 0.14%.

Ceiling. We draw a pool per domain with the audit's rules and size, without the duplicate filter of the original run. A pool paper whose community has n members in the pool can have at most $\min(10, n-1)$ of them among its top 10. The ceiling is the mean of $\min(10, n-1)/10$ over the pool: 97.9% at L1 and 70.6% at L2. Relative to it, the four embeddings reach 46–54% at L1 and 22–30% at L2.

E Protocols: fitted vector, held-out, fine-tuning and learned form

Fitted vector. For a set of documents we optimise k unit vectors $v_1, \ldots, v_k$. A document scores $\max_j \cos(v_j, x)$. The loss is the mean negative log-likelihood of the set's documents under a softmax over the whole corpus with temperature 0.05. We use Adam for 100 steps and initialise with the set's documents by farthest-point selection. The query paper and its citers are masked. The random control draws a set of the same size from the query's domain.

Fitted-vector results. For queries with more than 20 references, one vector fitted to the references finds .489 of them with bge-base and .600 with Gemini, and one vector fitted to a random set of the same size from the query's domain finds .083 and .145. Small sets are found easily, whether they are references or random. Reference sets are found better from six documents on with bge-base and from eleven documents on with Gemini, and the difference grows with set size (Table 13).

Table 13: R@100 of one vector fitted to a set, for the query paper's references and for a random set of the same size from the same domain, by set size.

Set size	3–5	6–10	11–20	21+
Reference set, bge-base	.972	.820	.652	.489
Random set, bge-base	.981	.760	.383	.083
Reference set, Gemini	.988	.861	.726	.600
Random set, Gemini	.998	.901	.582	.145

Held-out split. For each query with at least four references in the corpus, we shuffle the references and take the first half, rounded up, as A. Queries built from A are scored on B.

The query paper, its citers and A are masked when B is scored. The random control splits a random same-domain set of the same size in the same way. The MaxSim query scores a document by its maximum cosine to the documents of A, with or without the query paper.

Table 14: Held-out R@100 of half B of the references when half A is known and masked (queries with at least four references, no time constraint), so values differ from the full-gold R@100 elsewhere (bge-base .410 in Table 16), and bge-cite is bge-base fine-tuned on corpus citations. The last row fits a vector to half of a random same-domain set and scores the other half; † marks Gemini cells on a second random split.

Query	bge	bge-cite	Gemini
Query paper vector	.358	.386	.413
One vector fitted to A	.336	.375	.401
Four vectors fitted to A	.313	.338	.372
Documents of A as queries (MaxSim)	.268	.309	.335†
Query paper + A (MaxSim)	.336	.361	.407†
Random set: fit on half, score the other half	.000	.000	.000

†These cells use a second random split, on which the Gemini query paper vector scores .408; compare them with this value.

Table 15: Held-out gain of one vector fitted to the known half over the query paper's own vector (R@100, paired difference, 95% bootstrap CI), by the number of known references.

Known references	Queries	bge	bge-cite	Gemini
2	120	−.092 [−.163, −.017]	−.096 [−.175, −.021]	−.104 [−.175, −.033]
3–4	270	−.033 [−.071, +.006]	−.009 [−.048, +.029]	−.024 [−.063, +.016]
5 or more	231	+.029 [+.004, +.055]	+.030 [+.002, +.060]	+.050 [+.024, +.075]

Fine-tuning. We fine-tune bge-base-en-v1.5 (Xiao et al., 2024) with an in-batch contrastive loss on pairs of a paper and one of its references, batch size 512, learning rate 2×10^{-5} and one epoch. The held-out test uses the model trained on 4M pairs. Training papers have at least three references in the corpus, and each gives up to three of them. The 0.5M, 1M, 2M and 4M sets are nested random subsets of one pool, and bge-large-en-v1.5 is trained on the 2M set. Evaluation queries and papers that cite them are excluded from training. The time-split models use only the 472K training papers published up to 2018. The reference model uses 1.4M reference pairs. The hard-negative model adds, for each pair, one of the paper's 10th–100th base-model neighbours that it does not cite. The vote model pairs each paper with up to three documents that at least two of its 100 nearest base-model neighbours cite, without looking at its own references. The mixed model uses half reference pairs and half vote pairs. All four train for one epoch on about 1.4M pairs. We evaluate each encoder with its own embeddings of the corpus and the protocol of Section 3, with the query paper's vector as the query.

Table 16: Encoders fine-tuned on citations (R@100 with the query-paper vector, 95% bootstrap CI); training pairs are references, references with hard negatives, references and votes, or the neighbours' top-voted documents. Absorbed is the share of the base model's vote gain that fine-tuning gives to cosine alone; local share and vote minus popularity (on the vote's candidates) are in points.

Encoder	Training	Cosine	+ vote	Absorbed	Local share	Vote − pop.
No time constraint, 1,171 queries						
bge-base	–	.410	.533	–	+8.6 [+7.6, +9.6]	+4.7 [+3.5, +5.9]
bge-base	0.5M	.439	.547	24% [14%, 35%]	+6.8 [+5.9, +7.7]	+2.6 [+1.4, +3.8]
bge-base	1M	.448	.550	30% [20%, 42%]	+6.4 [+5.5, +7.2]	+2.0 [+0.8, +3.2]
bge-base	2M	.448	.548	31% [21%, 43%]	+6.2 [+5.3, +7.0]	+2.2 [+1.0, +3.3]
bge-base	4M	.447	.554	30% [19%, 42%]	+6.6 [+5.7, +7.4]	+2.4 [+1.2, +3.5]
bge-large	–	.411	.542	–	+9.2 [+8.1, +10.2]	+5.4 [+4.2, +6.6]
bge-large	2M	.459	.561	36% [27%, 47%]	+6.4 [+5.5, +7.3]	+2.5 [+1.4, +3.6]
Trained on papers up to 2018; 765 queries from 2019 on, time-constrained						
bge-base	–	.420	.516	–	+7.0 [+5.9, +8.2]	+5.3 [+4.1, +6.6]
bge-base	refs 1.4M	.458	.528	39% [23%, 56%]	+4.6 [+3.8, +5.5]	+2.6 [+1.6, +3.5]
bge-base	refs + hard neg.	.471	.534	53% [35%, 71%]	+4.5 [+3.6, +5.5]	+3.9 [+2.7, +5.2]
bge-base	refs + votes	.452	.517	33% [17%, 49%]	+4.3 [+3.5, +5.2]	+2.1 [+1.1, +3.1]
bge-base	votes only	.432	.501	12% [−4%, 28%]	+4.6 [+3.8, +5.5]	+2.0 [+1.0, +3.1]

Other text routes. For references outside the cosine top 1,000, exact match on named entities that an LLM extracts from the query abstract recovers 1.2% with a budget of 300 documents, and the embedding neighbours of the neighbours recover 0.8%.

Learned form. A histogram gradient-boosted classifier (300 iterations, 31 leaves, learning rate 0.1) scores each candidate from cosine, cosine rank, pool membership, and the logarithms of votes, popularity, co-citation and coupling counts, and personalised PageRank. It is trained pointwise with five-fold cross-validation grouped by query, on all positives and a random fifth of the negatives, and candidates are ranked by the predicted probability. On MasterSet and the time-constrained paper corpus, it reaches .392 and .553 with the text features only (cosine, cosine rank and pool membership), .638 and .690 with the text and citation features but without the vote, and .650 and .693 with all features. The features without the vote still include the other neighbourhood signals. With text and popularity only it reaches .532 and .613, and with text and the neighbourhood signals only (vote, co-citation, coupling and personalised PageRank) it reaches .648 and .694.

Citeomatic-style reranker. On the MasterSet candidates of the learned form, we build Citeomatic-style features (Bhagavatula et al., 2018), with TF-IDF cosines in place of its learned field embeddings: the first-stage cosine, cosine rank and pool membership, the TF-IDF cosine of the title and of the abstract, the sum of inverse document frequencies of the words that the query and the candidate share in the title and in the abstract, the number of shared authors, a same-venue indicator and the logarithm of the candidate's citation count. The learner and the cross-validation are those of the learned form. The model reaches .556, the neighbourhood signals alone .648, and the Citeomatic-style features with the neighbourhood signals .660. The paired differences in Section 5 are over the same 6,105 queries.

Query-vector updates. Dense pseudo-relevance feedback adds the mean of the query's top-10 cosine neighbours to the query vector. The citation Rocchio update adds the mean of the ten documents with the most votes from the query's top 100 neighbours other than its references. The multi-query variant uses the query paper and those ten documents as eleven queries and combines them by MaxSim or by reciprocal rank fusion. On bge-base with queries of at least three references, where the query vector scores .365, dense pseudo-relevance feedback reaches .357, the citation Rocchio update .358, and the multi-query variants .326–.332.

F Examples

Table 17: The four least-cited references of computer-science queries that fall outside the time-constrained cosine top 1,000 but enter the top 100 with the neighbour vote, after co-author neighbours are removed. Rank is the cosine rank, Votes the number of the query's 100 neighbours that cite the reference, Cites its in-corpus citations up to the query year, and one citing neighbour is shown.

Query	Reference	Rank	Votes	Cites	A citing neighbour
Audio-Visual Instance Discrimination with Cross-Modal Agreement	Split-Brain Autoencoders: Unsupervised Learning by Cross-Channel Prediction	2,637	4	62	Spatiotemporal Contrastive Video Representation Learning
Universal Video Style Transfer via Crystallization, Separation, and Blending	Combining Markov Random Fields and Convolutional Neural Networks for Image Synthesis	27,015	18	91	Real-Time Neural Style Transfer for Videos
Clinical concept and relation extraction using prompt-based machine reading comprehension	An overview of MetaMap: historical perspective and recent advances	1,905	8	135	Clinical Concept Linking with Contextualized Neural Representations
End-to-End Autonomous Driving: From Classic Paradigm to Large Model Empowerment—A Comprehensive Survey	Argoverse: 3D Tracking and Forecasting With Rich Maps	3,535	4	180	Parameterized Decision-Making with Multi-Modality Perception for Autonomous Driving

G Keyword front end

A common search pipeline first retrieves with keywords and then reranks. An LLM writes a short query from each query paper's title and abstract. BM25 (Robertson & Zaragoza, 2009) over the 4M corpus returns 1,000 candidates, with R@100 .237. Reranking the candidates by the Gemini embedding of the short query gives .356, and adding the neighbour vote, with the top 100 of these candidates as neighbours, gives .465. Dense retrieval over the whole corpus with the same short query gives .408. These numbers use no time constraint.