LOCKS AND KEYS: LEARNING SCIENTIFIC INSPIRATION RETRIEVAL AS PATTERN MATCHING AT SCALE

Anonymous authors
Paper under double-blind review

ABSTRACT

A new hypothesis often arises when a research problem meets complementary knowledge not previously associated with it. Making such discoveries thus begins with inspiration retrieval: finding, among the vast body of existing knowledge, the piece that fits the problem. The task looks unlearnable, since discoveries are new connections by definition and the ones a model must find at test time are absent from its training data. Yet it is learnable, and we propose why: inspiration retrieval is essentially pattern matching. A research problem is a lock and an inspiration is a key; individual pairs are new, but the way a problem's structure fits an inspiration's structure recurs, so the rule can be learned from records of past discoveries by ordinary supervised fine-tuning. Experiments bear this out: what a trained model learns is the fit, not a memory of similar pairs and not resemblance between problem and paper. The view also says how to learn the task: as a score on each problem–paper pair, computed by embedding and reranking models, rather than as a selection over a candidate set by a generative reasoning model, as in prior large-scale training. Given the same base model and annotations, scoring pairs is far more accurate than generative selection, at a fraction of the cost, and stays ahead as accuracy grows log-linearly over hundreds of thousands of training pairs. Finally, the learned fit uses only a small fraction of the embedding dimensions available, and the fraction does not increase with more training data: the task is limited not by representation capacity but by how much of the fit has been learned.

1 INTRODUCTION

Scientific discovery is often described as an act of connection: a new hypothesis arises when a research problem meets knowledge not previously associated with it. Recent work on LLM-driven discovery builds on this view (Luo et al., 2025). MOOSE-Chem (Yang et al., 2025) assumes that most hypotheses are compositions of a research background with a few inspirations, and shows that for such hypotheses discovery reduces to three sub-tasks: retrieving the inspirations, composing them with the background, and ranking the results. The assumption is supported both by cognitive science, which has long treated composition as a general mechanism of creative thought (Koestler, 1964; Benedek et al., 2012), and by empirical evidence: ResearchBench (Liu et al., 2026) finds that roughly 70–80% of papers in each of the twelve disciplines it tests can be decomposed in this way. Within this decomposition, inspiration retrieval is where the connection is made: given a background, find the piece of knowledge that, combined with it, yields the hypothesis. In practice, each such piece of knowledge is usually approximated by a paper that carries it.

This step raises an apparent paradox. If the connections that constitute discoveries are new by definition, then the pairs a retriever must recover at test time are absent from any corpus of past discoveries, and inspiration retrieval has accordingly been regarded as an out-of-distribution task (Yang et al., 2025; Liu et al., 2026). A growing line of work nonetheless treats it as a retrieval problem in its own right (Garikaparthi et al., 2025; Suzuki & Banaei-Kashani, 2026; Yang et al., 2026), and MOOSE-Star (Yang & Bing, 2026) recently showed that it can be trained at scale, using a generative reasoning model as the retriever and supervision decomposed automatically from scientific papers. These works establish that the task can be learned. What remains unexplained is what is being learned: which regularity, shared by training and test pairs, lets a model trained on past connections transfer to connections it has never seen—and, relatedly, how the task should be formulated so that this regularity is learned most directly.

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.

The paradox has a natural but, we argue, mistaken resolution: if the connections that make up discoveries are absent from every record of the past, then discovery cannot be learned from such records and must be learned from experience instead (Silver & Sutton, 2025). We propose a different resolution. Inspiration retrieval is essentially pattern matching. A research problem is a lock and an inspiration is a key: what the lock needs—a missing mechanism, an unmet constraint, a failing assumption—and what the key supplies are structural properties, and a key opens a lock when the two fit. A key need not resemble its lock, and it opens every lock cut to the same shape. A new problem is new in its content but not necessarily in its shape: what it lacks, and the kind of key that would supply it, recur across problems, and a model trained on enough past connections can learn which shapes call for which and apply the rule to problems it has never seen. Inspiration retrieval is out-of-distribution in the space of specific connections and in-distribution in the space of patterns, and so a central step of discovery is learnable by ordinary supervised fine-tuning on records of past discoveries, without learning from experience and without per-problem reasoning. We test this view directly. The gains of a trained model come neither from having seen pairs that look like the test pair, nor from resemblance between problem and paper: it retains most of its gains when every look-alike training pair is removed, and it still finds the inspiration when every candidate shares the problem's topic. What it learns is which problems call for the same kind of inspiration, and it recognizes this between problems that share no topic. That is what pattern matching predicts: a key need not look like its lock, and the same shape recurs across pairs that do not look alike.

The view also says how the task should be learned. If what transfers is the fit between a problem and an inspiration, the object to learn is a score on the pair, $s(q, d)$, that depends on the problem and the paper alone. We therefore learn a candidate-independent relational score, as a bi-encoder for corpus-scale search and a cross-encoder for reranking, both trained listwise and combined in a retrieve-then-rerank cascade. Existing methods fall on either side of this choice: dense retrievers fine-tuned on a few hundred problems in one domain (Garikaparthi et al., 2025), and set-conditioned selectors, which read a candidate set and reason their way to a choice, trained on over a hundred thousand decomposed examples (Yang & Bing, 2026). Only the latter has been tried at scale; the two have not been compared. Scoring pairs turns out to be both cheaper and more accurate. On the same base model and annotations as MOOSE-Star, top-1 accuracy rises from 54% to 72% with a bi-encoder and 84% with a cross-encoder; an off-the-shelf reranker matches the trained selector without fine-tuning. On the full corpus, mean rank of the true inspiration improves from 742 to 40, at one encoding per problem instead of dozens of generative calls, and the cascade matches the full cross-encoder at a fraction of its cost. The lead holds at every training scale, from zero-shot to hundreds of thousands of examples, and does not close; with base model and annotations fixed, it can only come from how the task is formulated and trained (§4.1). The gain carries through to the hypotheses: experts rank those generated from our inspirations close to those from the annotated inspiration and well ahead of those from MOOSE-Star's.

If the task is pattern matching learned as a score on pairs, a natural question is what limits it: the capacity of the embedding space or the amount of training data. Theory offers a bound on the first: a retrieval relation in which each problem has at most k inspirations is representable in a number of dimensions that depends on k alone, not on how many problems and papers it contains (Wang et al., 2026). We bring this bound to inspiration retrieval for the first time and test it: across four backbones, the learned fit reaches nearly all of its full-width accuracy in a small fraction of the dimensions the models provide, and needs no more of them as the training set grows. Capacity is in surplus. What keeps growing is accuracy with data: it rises roughly log-linearly for bi-encoders, cross-encoders, and set-conditioned selectors alike, with no sign of saturation. At the scales we study, inspiration retrieval is limited not by the dimension of its embedding space but by how much of the fit has been learned.

Our contributions are threefold.

- **Why the task is learnable.** We propose that inspiration retrieval is pattern matching between the structure of a research problem and that of an inspiration, so that a central step of discovery can be learned by supervised fine-tuning on records of past discoveries, and we confirm the view experimentally.

- **A better way to learn it.** Learning a candidate-independent score on pairs, as a bi-encoder, a cross-encoder, or their cascade, is far more accurate than MOOSE-Star's set-conditioned

selector on the same base model and annotations, at a fraction of the cost, at every training scale, and the gain carries through to the quality of the resulting hypotheses.

- **What limits it.** Testing embedding dimension and training data as the two candidate bottlenecks, we find that dimension is in surplus and data is what limits the task.

2 Related Work

LLM systems generate hypotheses grounded in the literature, by retrieving and refining toward novelty (Wang et al., 2024), modular feedback (Yang et al., 2024), or citation graphs with iterative review (Baek et al., 2025); MOOSE-Chem and ResearchBench decompose discovery into inspiration retrieval, composition, and ranking (Yang et al., 2025; Liu et al., 2026), and MOOSE-Star trains the decomposed process at scale on supervision derived from papers (Yang & Bing, 2026). In all of them retrieval is one stage of a pipeline. Retrieval itself has moved past topic matching: SPECTER and ASPIRE learn from citation signals (Cohan et al., 2020; Mysore et al., 2022), LitSearch and BRIGHT pose queries whose relevant documents take reasoning to find (Ajith et al., 2024; Su et al., 2025), ReasonIR trains on such queries (Shao et al., 2025), and Promptriever follows instructions (Weller et al., 2025), though instructed retrievers can still miss the aspect a scientist asks for (Maheshwari et al., 2026). All retain the objective of finding documents that satisfy an information need, whereas an inspiration supplies what a problem lacks and need not resemble it. Methods built for this task add structure to the retriever—a citation-derived methodology graph (Garikaparthi et al., 2025), per-pair transferability abstractions (Suzuki & Banaei-Kashani, 2026), hyperbolic bridge modeling (Yang et al., 2026)—or, in MOOSE-Star, replace it with a generative selector that reads a candidate set and reasons to a choice (Yang & Bing, 2026). We instead learn a candidate-independent score on the problem–paper pair, which admits a bi-encoder for corpus search and a cross-encoder for reranking, and we ask why such a score can be learned at all.

3 Method

Our approach rests on one modeling decision: the fit between a research problem and an inspiration is learned as a score on the pair, independent of the other candidates present. We motivate this decision from a pattern-matching view of the task (§3.1), realize the score with a bi-encoder for corpus-scale retrieval and a cross-encoder for reranking, both trained listwise (§3.2), and chain the two into a retrieve-then-rerank cascade (§3.3). Figure 1 summarizes the pipeline.

3.1 Inspiration Retrieval as Pattern Matching

Setting. We follow the setting of MOOSE-Star (Yang & Bing, 2026), in which a hypothesis is built in steps, each step adding one inspiration. The *research problem* at step j is $q_j = (r, b, h_{j-1})$: the research question r, a background survey b of existing approaches and their limitations, and, after the first step, a description h_{j-1} of the approach assembled so far from earlier inspirations ($h_0 = \varnothing$). For retrieval, h_{j-1} plays the same role as b: together they are what is already known, and r is what is asked. An *inspiration* is a piece of knowledge that is complementary to what is known and not yet known to be related to r; combined with what is known, it yields the next step of the hypothesis. Knowledge cannot be retrieved directly. It is found in papers, roughly one piece per paper, so following prior work we approximate an inspiration by the paper $d = (t, a)$ that carries it, given by title and abstract, and retrieval ranges over a corpus $\mathcal{D}$ of papers. For each research problem the benchmark annotates one inspiration d_j^+ and pairs it with a set $\mathcal{C}_{q_j}$ of candidate papers, containing d_j^+ and other papers, from which the inspiration is to be picked out.

Locks and keys. An inspiration is defined not by what it is but by what it does: it supplies what the problem lacks. A problem states a need. Its background records where existing approaches stop and why, its question what is wanted, and between them they say what is missing—a mechanism, a constraint no current method meets, an assumption that has to go. A paper states a supply: the mechanism it introduces, the resource it provides, the reframing it makes possible. Whether d inspires q is whether the supply meets the need—a relation between the two sides, not a resemblance between them. An inspiring paper usually comes from another field and shares little vocabulary with the problem; what it shares is that it fits. We call the need and the supply the *structures* of the

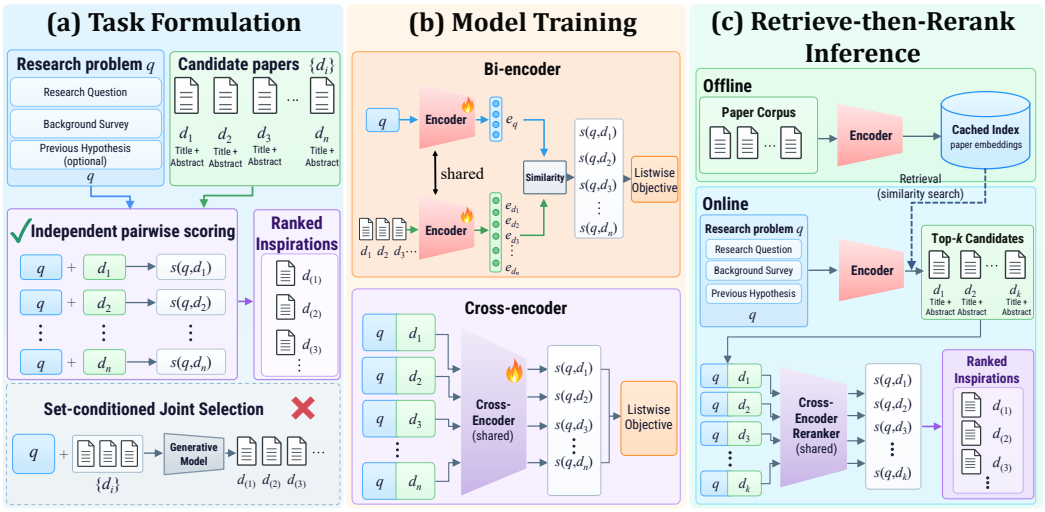


Figure 1: (a) Each candidate paper is scored on its own against the research problem, rather than selected jointly from a set. (b) Both the bi-encoder and the cross-encoder are fine-tuned with a listwise objective. (c) Cached paper vectors support corpus-scale retrieval; the cross-encoder reranks only the shortlist.

problem and of the paper, and we picture their relation as a lock and a key. A key does not resemble the lock it opens. It opens it because the shapes fit, and for the same reason it opens every lock cut to that shape. Our hypothesis is that inspiration retrieval is *pattern matching* in exactly this sense: whether d fits q is decided by their structures, through one rule that holds across pairs. This is why a task that is new by construction can nonetheless be learned. The connections a retriever is asked to find are discoveries, absent from any record of the past, so in the space of pairs the test set is out of distribution. But if the rule recurs, the structures it connects have appeared before, and in the space of structures the same test set is in distribution: the rule can be learned from past pairs and applied to pairs never seen. §4.2 tests this hypothesis.

The fit as a score on pairs. If fit is decided by the structures of the two sides, then it is a property of the pair, and the object to learn is a score on pairs, $s_\theta : \mathcal{Q} \times \mathcal{D} \to \mathbb{R}$, increasing in how well d fits q; retrieval selects $\hat{d}_j = \arg\max_{d \in \mathcal{C}_{q_j}} s_\theta(q_j, d)$. The defining property of such a score is *candidate independence*: $s_\theta(q, d)$ depends on q and d alone, just as whether a key fits a lock does not depend on the other keys on the ring, so changing the candidate set changes ranks but never scores. This separates our formulation from *set-conditioned selection*, $\hat{d}_j = g_\phi(q_j, \mathcal{C}_{q_j})$, the formulation behind MOOSE-Star's generative selector, in which the decision about a paper may depend on the identity, order, and number of the papers shown with it, and every candidate must be re-read for every problem. Candidate independence is also what makes corpus-scale search possible, since paper representations can be computed once and reused (§3.3).

3.2 Learning the Fit: Bi-encoder and Cross-encoder

Two architectures instantiate s_θ; they differ in how the score is computed and share the objective. Below we write q for a problem, d^+ for its annotated inspiration and $\mathcal{C}_q$ for its candidate set, dropping the step index j.

Bi-encoder. This is the lock-and-key picture taken literally: one map cuts the problem, one cuts the paper, and the dot product tests the fit. A shared encoder f_θ with last-token pooling maps the serialized problem $x(q)$ and paper $y(d)$ to unit vectors u_q and v_d, and

$$s_{\mathrm{B}}(q, d) = u_q^\top v_d. \tag{1}$$

Because v_d does not depend on q, papers are encoded once and reused across problems. What fine-tuning must learn is not how to embed a problem—the pretrained encoder already does that—but

which regions of problem space are compatible with which regions of paper space, and that can be learned only from pairs.

Cross-encoder. This tests the fit by reading the two sides together. A decoder backbone takes problem and paper through a template $T(q, d)$ and returns a scalar from its final-token logits,

$$s_{\mathrm{C}}(q, d) = F_{\psi}\big(T(q, d)\big), \tag{2}$$

so the representation of the paper can attend to the particular question, limitation, and progress so far. No candidate other than d enters $T(q, d)$.

Objective. Both models are fine-tuned to rank d^{+} above the other candidates, which we call *negatives*, with a softmax over the candidate set:

$$\mathcal{L}(q) = -\log \frac{\exp\big(s(q, d^{+})/\tau\big)}{\sum_{d' \in \mathcal{C}_q} \exp\big(s(q, d')/\tau\big)} \tag{3}$$

(for the bi-encoder the denominator also includes the papers of the other problems in the batch, as in InfoNCE, van den Oord et al., 2018). We also train a pointwise variant, which replaces this softmax with a binary loss on each pair, positive for d^{+} and negative for the rest; §4.1 compares it with the listwise objective above. The loss compares scores across candidates, but each score is a function of one pair: competition enters through the loss, never through the model, so training has what a selection objective has while inference keeps candidate independence.

3.3 Retrieve, then Rerank

The two scores have complementary strengths. The bi-encoder never sees the problem and the paper together, but its paper vectors can be computed once and searched over an entire corpus; the cross-encoder sees both sides and scores more accurately, but must run a full forward pass for every pair. We therefore chain them: the bi-encoder searches the corpus, and the cross-encoder decides among the few papers it returns.

Concretely, the vectors v_d of all papers in the search collection $\mathcal{D}_{\mathrm{search}}$ are computed once by the bi-encoder and cached. For a new problem q, the bi-encoder encodes q and retrieves the K papers with the highest $s_{\mathrm{B}}(q, d)$, the shortlist $\mathcal{R}_K(q)$; the cross-encoder then scores each pair (q, d), $d \in \mathcal{R}_K(q)$, and returns $\hat{d} = \arg\max_{d \in \mathcal{R}_K(q)} s_{\mathrm{C}}(q, d)$. Both stages score pairs; the collection affects only which papers reach the second stage. The online cost per problem is one encoding, one vector search, and K cross-encoder passes instead of $|\mathcal{D}_{\mathrm{search}}|$, so K sets the trade-off between cost and recall: the cross-encoder can only reorder what the bi-encoder returns.

4 Experiments

Inspiration retrieval is out of distribution by construction: every test pair is a discovery, absent from any earlier record. The task can be learned only if training and test pairs share a regularity, and §3.1 named one: what a paper supplies fits what a problem lacks, by a criterion that recurs across pairs. We ask three questions. Is the fit learnable as a score on pairs, how well, and at what cost, against selecting from the set (§4.1)? Which regularity did the learner find: the training pairs themselves, a property of the paper alone, a resemblance between the two sides, or the fit (§4.2)? And how much capacity does the regularity need: how many dimensions, and whether accuracy is limited by the width of the model or by the amount of training data (§4.3)?

Setup. We evaluate scientific inspiration retrieval on TOMATO-Star (Yang & Bing, 2026): given a research problem (a research question with background), the task is to rank candidate papers by their potential as scientific inspirations. In the *Original* setting, each query is paired with one annotated inspiration and 14 negatives from a mixture of random and similarity-based sampling; we also evaluate three controlled candidate distributions (*All Random*, *All Hard*, and *Decoy Cluster*) that vary negative difficulty and candidate-set composition. We compare the set-conditioned generative selector MS-IR-7B (Yang & Bing, 2026) with bi-encoder retrievers (Qwen3-Embedding 0.6B/4B/8B

(Zhang et al., 2025) and GTE-Qwen2-7B-Instruct (Li et al., 2023)) and cross-encoder rerankers (Qwen3-Reranker 0.6B/4B/8B (Zhang et al., 2025) and BGE-Reranker-v2-Gemma[1]).

4.1 Scoring Pairs versus Set-Conditioned Selection

Main comparison. Scoring pairs beats selecting from the set by a wide margin and with far smaller models (Table 1). MS-IR-7B, the set-conditioned selector, reaches 54.4% top-1 on Original; every fine-tuned pair scorer exceeds it, from 74.4% (GTE-Qwen2-7B) to 86.3% (Qwen3-Reranker-8B), and the 0.6B reranker reaches 81.1% with 11.7× fewer parameters. Retrieve-then-rerank cascades match or exceed their rerankers. On the corpus, the mean rank of the annotated inspiration falls from 742 to 32–50. The lead holds on every candidate construction—All Random, All Hard, Decoy Cluster (Tables 9–10)—and without the background field (Table 11), and at every training size we test, from 10^2 to 2×10^5 pairs (Fig. 4).

Table 1: Top-1 accuracy (%) on the four candidate distributions and mean rank (Proposed Rank) over a pool of 3,035 candidates. FT: Fine-Tuned.

Models	Original	All Random	All Hard	Decoy Cluster	Proposed Rank↓
MS-IR-7B (Yang & Bing, 2026)	54.37	78.58	53.42	77.65	742.50
Qwen3-Embedding-0.6B (Zhang et al., 2025)	35.09	73.15	37.77	73.85	192.53
Qwen3-Embedding-0.6B-FT	77.39	88.31	79.12	90.77	49.44
Qwen3-Embedding-4B (Zhang et al., 2025)	37.61	76.79	43.20	78.77	146.48
Qwen3-Embedding-4B-FT	82.12	90.20	84.64	92.59	41.18
Qwen3-Embedding-8B (Zhang et al., 2025)	38.38	79.50	45.08	80.75	128.64
Qwen3-Embedding-8B-FT	82.50	90.49	85.28	93.23	40.13
gte-Qwen2-7B-instruct (Li et al., 2023)	38.28	73.56	45.69	75.99	172.08
gte-Qwen2-7B-instruct-FT	74.43	84.55	77.04	88.03	92.15
Qwen3-Reranker-0.6B (Zhang et al., 2025)	43.14	79.28	45.21	80.59	139.18
Qwen3-Reranker-0.6B-FT	81.13	87.20	83.62	89.97	50.12
Qwen3-Reranker-4B (Zhang et al., 2025)	52.71	85.86	55.36	86.46	80.70
Qwen3-Reranker-4B-FT	85.63	91.16	87.80	93.01	34.59
Qwen3-Reranker-8B (Zhang et al., 2025)	52.97	84.42	56.10	85.44	92.35
Qwen3-Reranker-8B-FT	86.33	91.41	88.31	93.52	31.85
BGE-Reranker-v2-Gemma[1]	42.43	76.60	44.28	77.04	163.80
BGE-Reranker-v2-Gemma-FT	82.38	88.41	84.74	90.71	45.75
Qwen3-Embedding-0.6B-FT (Top-3)→Qwen3-Reranker-0.6B-FT	81.29	87.84	83.30	90.17	49.48
Qwen3-Embedding-4B-FT (Top-3)→Qwen3-Reranker-4B-FT	86.27	91.22	88.09	93.01	41.19
Qwen3-Embedding-8B-FT (Top-3)→Qwen3-Reranker-8B-FT	86.49	91.70	88.22	93.55	40.14
gte-Qwen2-7B-instruct-FT (Top-3)→BGE-Reranker-v2-Gemma-FT	81.80	87.52	84.36	89.91	92.14

Why scoring pairs wins. The advantage is the formulation, not the backbone. On MS-IR-7B's own backbone, DeepSeek-R1-Distill-Qwen-7B, a bi-encoder and a cross-encoder fine-tuned from the same annotations reach 71.6% and 83.5% against the selector's 54.4% (Table 2). A pair score is trained directly on the annotated fit, with competition entering only through the loss (§3.2); a selector must re-derive the fit inside a reasoning trace that also depends on the other candidates shown. The listwise objective helps on its own, beating pointwise on every backbone (Table 17).

Table 2: Top-1 accuracy (%) on the four candidate distributions and mean rank (Proposed Rank) over a pool of 3,035 candidates. MS-IR-7B is fine-tuned from DeepSeek-R1-Distill-Qwen-7B, which is also adapted into an embedding model (DeepSeek-R1-Distill-Qwen-7B-Embedding) and a reranking model (DeepSeek-R1-Distill-Qwen-7B-Reranker) by us. FT: Fine-Tuned.

Models	Original	All Random	All Hard	Decoy Cluster	Proposed Rank↓
MS-IR-7B (Yang & Bing, 2026)	54.37	78.58	53.42	77.65	742.50
DeepSeek-R1-Distill-Qwen-7B-Embedding (Guo et al., 2025)	8.68	10.50	8.75	10.03	1409.69
DeepSeek-R1-Distill-Qwen-7B-Embedding-FT	71.58	89.56	73.18	91.09	45.42
DeepSeek-R1-Distill-Qwen-7B-Reranker (Guo et al., 2025)	10.60	12.45	9.10	11.02	1396.62
DeepSeek-R1-Distill-Qwen-7B-Reranker-FT	83.52	89.21	85.22	91.19	41.89

[1] https://huggingface.co/BAAI/bge-reranker-v2-gemma

At a fraction of the cost. Selecting from the set re-reads every candidate and decodes a reasoning trace; scoring pairs encodes the problem once, with no decoding. On the same backbone (Table 21), MS-IR-7B takes 24.7 s per problem, the fine-tuned bi-encoder 18,ms at 55× fewer FLOPs, and the top-3 cascade 0.1 s at the cross-encoder's accuracy (Table 19).

From inspirations to hypotheses. The gain carries through to the hypotheses. With the generator fixed, an LLM judge ranks hypotheses built from our inspirations at 1.93 among four sources (random 3.39, MS-IR 2.72, annotated 1.71); three experts on 50 problems rank them at 2.11, on par with the annotated inspiration (2.09) and ahead of MS-IR (2.33) (Table 3, Appendix E).

Table 3: Downstream hypothesis quality with different retrieval methods. The LLM-based evaluation covers the full test set, while the expert evaluation uses a random sample of 50 instances.

Method	LLM Mean Rank↓	Expert Mean Rank↓			
		Expert 1	Expert 2	Expert 3	Average
Random	3.39	3.36	3.10	2.82	3.09
MS-IR-7B (Yang & Bing, 2026)	2.72	2.52	2.16	2.30	2.33
Ours (Retrieve→Rerank)	1.93	2.04	2.04	2.24	2.11
Ground Truth	1.71	1.70	2.32	2.26	2.09

4.2 Is Inspiration Retrieval Learnable Pattern Matching?

All analyses in this section use Qwen3-Embedding-0.6B; training recipes are in Table 5.

The rule is general. The training pairs themselves are not what transfers. We delete from the training set every pair that resembles a test pair—problem and inspiration both close to some test problem, 34% of the rows—and retrain: recovery (the share of the base model's 2,033 failures a model fixes) falls only from 65.7% to 63.4% relative to deleting the same number of rows at random, so 96% of it survives (Fig. 8). Deleting rows that resemble a test problem but carry a dissimilar inspiration costs nothing (65.9%), and a model trained on 2k random rows still recovers 55% on problems those rows leave uncovered. The recovery gradient over coverage is mostly difficulty: after deletion only the top bins move.

What is matched is the relation. Nor is it a property of the paper alone or a resemblance between the two sides (Table 22). Training on the same papers with their pairing scrambled drives top-1 to 12.2%, below the base model's 35.1%: nothing useful is learned about papers or format without the pairing. With every candidate on the problem's own topic (All Hard), the fine-tuned model still finds the inspiration 79.1% of the time (base 37.8%); BM25 scores 30.9%, and problems share almost no 4-grams with their inspirations. On multi-step problems, the base model re-selects a paper already used in the hypothesis 44.9% of the time, the fine-tuned model 23.4%: it is not matching text against the hypothesis but supplying what it still lacks.

What the learned similarity is. What remains is the fit, and the learned geometry shows it. For pairs of test problems from different papers, similarity under the fine-tuned encoder predicts how similar their inspirations are (0.29) where similarity under the base encoder barely does (0.07), and it does so after controlling for base similarity (partial Spearman 0.43; Fig. 6). A blind LLM judge on nearest-neighbour pairs sees the same: fine-tuning halves the share that share a topic (59.7% to 38.3%) while leaving the share that share a problem structure unchanged (17.1% to 17.3%), so pairs alike in structure but not in topic nearly double (5.3% to 9.2%; Fig. 2). The similarity the model learns is distinct from topic and consistent with the structural reading of §3.1.

The local component, and caveats. What the look-alike pairs do carry is small and neighbourhood-level. Removing each test problem's 30 nearest training rows in the learned space costs 1.5 points; adding 2k rows chosen for joint similarity to test pairs raises recovery by 5–8 points over 2k random rows, equally on held-in and held-out halves of the test set, and the best rule copying neighbours' inspirations reaches 16.9% on the corpus against the model's 23.5% (Fig. 7). Coverage changes which paper is chosen, not how useful: a hypothesis-quality judge is flat across coverage. Caveats: one bi-encoder; one or two seeds for deletion runs; LLM rather than expert labels.

(a) Neighbour semantics

Label	Base	FT	FT - Base
All neighbours			
Topic = yes	**48.6** [45.1,52.2]	**29.3** [26.1,32.5]	**-19.3** [-22.4,-16.4]
Structure = yes	**4.4** [3.5,5.4]	**6.9** [5.5,8.3]	**+2.5** [1.2,3.8]
Among topic ≠ yes			
Structure = yes	**6.4** [5.0,7.9]	**8.5** [6.8,10.4]	**+2.1** [0.3,4.0]
Structure ≥ partly	**38.6** [35.0,42.3]	**45.2** [41.8,48.8]	**+6.6** [2.8,10.6]

Rates: %; differences: percentage points.
95% CIs: question-level bootstrap.
300 questions; 10 neighbours / encoder.

(b) FT neighbour: example 2

LLM judge: topic = no; structure = yes

Test | PD-MCI. How to detect early-stage cognitive decline in Parkinson's disease using electrophysiological biomarkers ... with high diagnostic accuracy?

Key: *Beyond broadband: Towards a spectral decomposition of electroencephalography microstates.*

Neighbour | Early AD. How to develop a low-cost, non-invasive functional biomarker that detects early-stage Alzheimer's disease by capturing impaired brain network dynamics ...?

Key: *EEG microstates as a tool for studying the temporal dynamics of whole-brain neuronal networks: A review.*

Judge: "existing clinical and EEG methods lack sensitivity/specificity ... because they assume stationarity and miss transient network dynamics, requiring a low-cost electrophysiological biomarker capturing transient, frequency-specific network alterations."

Figure 2: Blind LLM judgments (glm-5.3) of whether a test problem and its ten nearest training problems share a topic or a problem structure, under the base and the fine-tuned encoder; 300 problems, bootstrap 95% intervals. Fine-tuning removes topical overlap and preserves structural overlap. Right: a fine-tuned-space neighbour pair judged different topic, same structure.

4.3 Capacity, Dimension, and Data

What theory says. Let $Y \in \{-1, +1\}^{N \times M}$ encode a problem–paper relation and let its *sign-rank* be the smallest rank of a real matrix with sign pattern Y. Weller et al. (2026) bound the embedding dimension $d_{\min}$ that ranks every positive above every negative for each query by $\mathrm{signrank}(Y) - 1 \leq d_{\min} \leq \mathrm{signrank}(Y)$, so arbitrary relations can require growing dimension, and realizing every size-k answer set with a fixed score gap requires dimension growing with corpus size. Without the gap, Wang et al. (2026) show by a cyclic-polytope construction that every answer set of at most k papers is separable in $2k$ dimensions under inner product and $2k + 1$ under cosine, independently of corpus size. With one annotated inspiration per problem, two dimensions suffice to represent the annotated relation. This is an existence statement for free vectors; it says nothing about learnability, and the annotated relation is not the true relation.

How much dimension the learned fit uses. Far less than the models provide. Fine-tuned at 32 dimensions, the four bi-encoders reach about 97% of their full-width top-1 on candidate sets, and within one point at 128–512 (Fig. 3); on the 3,035-paper corpus, 128 dimensions retain 93–96% of full-width MRR while the base model keeps improving to full width, and an 8-dimensional fine-tuned model already outranks the full-width base (Appendix D). The theoretical floor is not reached by training: a two-dimensional model fits neither its training set nor the test set (14.2% ± 0.1 over three seeds, below the base model's 35.1%).

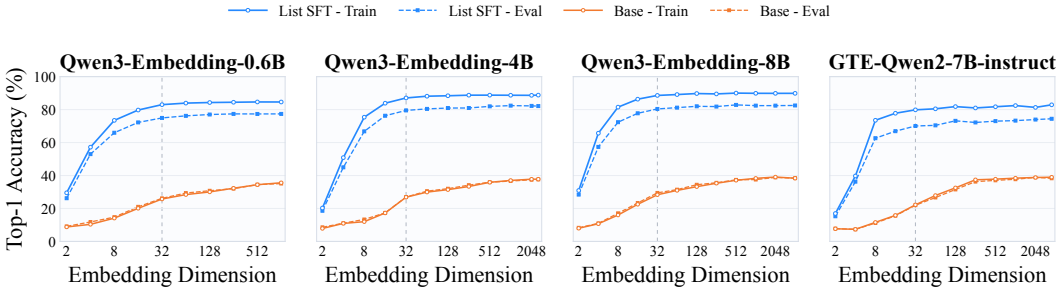


Figure 3: Effect of embedding dimension on top-1 accuracy (%) for four embedding models, before and after listwise supervised full fine-tuning, evaluated on the training set and the Original test set.

Table 4: Original top-1 accuracy (%) of Qwen3-Embedding-0.6B across training-set sizes and embedding dimensions. Retention is the accuracy at $d = 128$ relative to $d = 1024$. Training recipes differ across rows; comparisons are within-row (see Appendix B.1 for settings).

Training pairs	$d = 2$	$d = 8$	$d = 32$	$d = 128$	$d = 1024$	Retained (%)
1,000	13.22	30.40	58.43	63.35	63.06	100.5
10,000	14.24	57.12	66.83	69.28	70.66	98.1
195,301	26.21	65.96	74.94	77.01	77.39	99.5

Dimension versus data. The dimension the fit needs does not grow with the number of training pairs. At 1k, 10k, and 195k pairs, 128 dimensions retain 98–100% of the accuracy at 1,024 (Table 4); what more data buys is accuracy at every width (at 32 dimensions, 58.4% to 66.8% to 74.9%), and at two dimensions it never lifts the model above the base. Rows use different recipes, so only within-row trends are compared.

What limits the task. Data, not width. Beyond a few hundred dimensions, width buys nothing at any training size, while accuracy keeps rising roughly log-linearly with training pairs for every model family, with 4B and 8B nearly tied (Fig. 4). What is encoded still matters: cross-encoders, which read the pair together, lead bi-encoders by 3.5–12 points (Table 1). At the scales we study, inspiration retrieval is limited by how much of the fit has been learned.

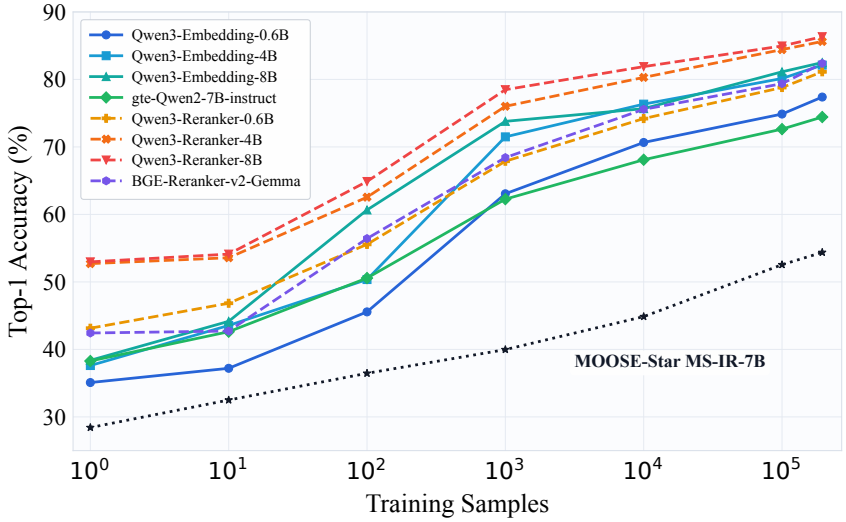


Figure 4: Scaling laws of embedding and reranking models on the Original test set.

5 Conclusion

Inspiration retrieval looks unlearnable, since every discovery is a connection absent from the record; we have argued that it is learnable because it is pattern matching. A research problem is a lock and an inspiration is a key, and although each pair is new, which shapes call for which recurs. A model trained on records of past discoveries learns this rule: its gains survive the removal of every training pair that resembles a test pair, do not rest on a property of the paper or on resemblance between problem and paper, and reflect a similarity between problems that is distinct from topic. The rule is best learned as it is stated, as a score on the pair: bi-encoders and cross-encoders trained on the same annotations are far more accurate than a generative selector at a fraction of its cost, at every training scale, and the better inspirations yield better hypotheses. The fit they learn occupies a small fraction of their embedding dimensions, no more as data grows, while accuracy keeps rising with data. A central step of discovery can thus be learned by ordinary supervised fine-tuning, and what it needs is more records of past discoveries, not more capacity.

AI USE STATEMENT

In this work, we used generative AI tools for [translation; provide feedback on research methodology or experiments; implementing methods; supporting qualitative and thematic data analysis; generating synthetic datasets; interpreting results; providing critical ingredients for mathematical proofs; cleaning and reformatting datasets; assisting with proof writing; proposing or refining hypotheses]. We have not used generative AI tools for [formulating mathematical claims; developing theoretical models or conceptual frameworks], Additionally, we used generative AI tools for [drafting and revising parts of the manuscript; improving readability and organization; identifying and reviewing relevant literature; searching for information; checking and formatting references]. We have reviewed all AI-assisted work. [Specifically, the authors checked AI-assisted translations and manuscript revisions for fidelity and technical accuracy, verified references against their original sources, and reviewed AI-assisted implementations for consistency with the described methods. We also checked AI-assisted qualitative analyses and interpretations against the underlying research materials and experimental results.]. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies scientific inspiration retrieval using existing scientific literature and benchmarks. Retrieval systems may reflect biases and gaps in the underlying literature and annotations. Retrieved papers and generated hypotheses should therefore be treated as research aids rather than validated scientific conclusions. Their use requires appropriate attribution, expert judgment, and independent verification, including experimental validation where applicable.

REPRODUCIBILITY STATEMENT

To support reproducibility, Section 3 describes the task formulation, model scoring functions, training objectives, and retrieve-then-rerank procedure. Section 4 specifies the benchmark, candidate-set settings, model backbones, and primary evaluation metric. Additional experimental results and ablation studies are reported in the appendix.

REFERENCES

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. doi: 10.18653/v1/2024.emnlp-main.840.

Jinheon Baek, Sujay Kumar Jauhar, Silviu Cucerzan, and Sung Ju Hwang. ResearchAgent: Iterative research idea generation over scientific literature with large language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics*, pp. 6709–6738, 2025. doi: 10.18653/v1/2025.naacl-long.342.

Mathias Benedek, Tanja Könen, and Aljoscha C Neubauer. Associative abilities underlying creativity. *Psychology of Aesthetics, Creativity, and the Arts*, 6(3):273, 2012.

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. doi: 10.18653/v1/2020.acl-main.207.

Aniketh Garikaparthi, Manasi Patwardhan, Aditya Sanjiv Kanade, Aman Hassan, Lovekesh Vig, and Arman Cohan. MIR: Methodology inspiration retrieval for scientific research problems. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, pp. 28614–28659, 2025. doi: 10.18653/v1/2025.acl-long.1390.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Arthur Koestler. The act of creation. *London: Hutchinson*, 1964.

Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281*, 2023. URL https://arxiv.org/abs/2308.03281.

Yujie Liu, Zonglin Yang, Tong Xie, Jinjie Ni, Ben Gao, Yuqiang Li, Shixiang Tang, Wanli Ouyang, Erik Cambria, and Dongzhan Zhou. ResearchBench: Benchmarking LLMs in scientific discovery via inspiration-based task decomposition. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 13187–13207, 2026. doi: 10.18653/v1/2026.findings-acl.644.

Ziming Luo, Zonglin Yang, Zexin Xu, Wei Yang, and Xinya Du. Llm4sr: A survey on large language models for scientific research. *arXiv preprint arXiv:2501.04306*, 2025.

Piyush Maheshwari, Sheshera Mysore, and Hamed Zamani. Can instructed retrieval models really support exploration? In *Proceedings of the 2026 Conference on Human Information Interaction and Retrieval*, pp. 508–513, 2026. doi: 10.1145/3786304.3787888.

Sheshera Mysore, Arman Cohan, and Tom Hope. Multi-vector models with textual guidance for fine-grained scientific document similarity. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 4453–4470, 2022. doi: 10.18653/v1/2022.naacl-main.331.

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*, 2025. URL https://arxiv.org/abs/2504.20595.

David Silver and Richard S Sutton. Welcome to the era of experience. *Google AI*, 1(11):6, 2025.

Hongjin Su, Howard Yen, Mengzhou Xia, Weijia Shi, Niklas Muennighoff, Han-yu Wang, Haisu Liu, Quan Shi, Zachary S. Siegel, Michael Tang, Ruoxi Sun, Jinsung Yoon, Sercan O. Arik, Danqi Chen, and Tao Yu. BRIGHT: A realistic and challenging benchmark for reasoning-intensive retrieval. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=ykuc5q381b.

Yuto Suzuki and Farnoush Banaei-Kashani. TCA-SIR: Learning target-conditioned abstractions for scientific inspiration retrieval. *arXiv preprint arXiv:2607.28498*, 2026. URL https://arxiv.org/abs/2607.28498.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. URL https://arxiv.org/abs/1807.03748.

Qingyun Wang, Doug Downey, Heng Ji, and Tom Hope. SciMON: Scientific inspiration machines optimized for novelty. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pp. 279–299, 2024. doi: 10.18653/v1/2024.acl-long.18.

Zifeng Wang, Ziqi Yin, Rui Liu, Junlin Tong, Ziqiao Song, Weng Lam Wong, and Simon See. $\mathbb{R}^{2k}$ is theoretically large enough for embedding-based top-k retrieval. In *International Conference on Machine Learning (ICML)*, 2026.

Orion Weller, Benjamin Van Durme, Dawn Lawrie, Ashwin Paranjape, Yuhao Zhang, and Jack Hessel. Promptriever: Instruction-trained retrievers can be prompted like language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=odvSjn416y.

Orion Weller, Michael Boratko, Iftekhar Naim, and Jinhyuk Lee. On the theoretical limitations of embedding-based retrieval. In *International Conference on Learning Representations*, volume 2026, pp. 11202–11216, 2026.

Yang Yang, Boyun Xu, Hao Fu, Jindong Li, Zining Zhong, Bowen Tian, Jiemin Wu, Menglin Yang, and Yutao Yue. HyBIRD: Hyperbolic bridge retrieval and diagnosis for methodology inspiration retrieval. *arXiv preprint arXiv:2606.28336*, 2026. URL https://arxiv.org/abs/2606.28336.

Zonglin Yang and Lidong Bing. MOOSE-Star: Unlocking tractable training for scientific discovery by breaking the complexity barrier. In *The 43rd International Conference on Machine Learning*, 2026. URL https://icml.cc/virtual/2026/poster/64492.

Zonglin Yang, Xinya Du, Junxian Li, Jie Zheng, Soujanya Poria, and Erik Cambria. Large language models for automated open-domain scientific hypotheses discovery. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 13545–13565, 2024. doi: 10.18653/v1/2024.findings-acl.804.

Zonglin Yang, Wanhao Liu, Ben Gao, Tong Xie, Yuqiang Li, Wanli Ouyang, Soujanya Poria, Erik Cambria, and Dongzhan Zhou. MOOSE-Chem: Large language models for rediscovering unseen chemistry scientific hypotheses. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=X9OfMNNepI.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. Qwen3 Embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025. URL https://arxiv.org/abs/2506.05176.

A Data and Experimental Details

A.1 Dataset Construction and Temporal Splits

We use TOMATO-Star and the temporal split described in the main text: source papers from January 2020 through September 2025 form the training period, and October 2025 papers form the test period. The main evaluation contains 3,132 queries. The additional setting with 150,218 training examples and 2,377 test queries is reported separately in Table 13; its scores should not be treated as evaluations on the same test set.

A.2 Retrieval Contexts and Candidate Pool Construction

A retrieval context contains a research question and background, with the available hypothesis state included where applicable. Candidates are represented by titles and abstracts. Original contains one annotated inspiration and 14 negatives. All Random, All Hard, and Decoy Cluster use random negatives, lexically or semantically related negatives, and a coherent but irrelevant cluster associated with another query, respectively. The All-Positive pool contains 3,035 annotated inspiration papers from the test set. Its name describes the origin of the pool, not that every paper is relevant to every query.

A.3 Model Backbones and Input Templates

The primary families are Qwen3-Embedding, Qwen3-Reranker, GTE-Qwen2-7B-Instruct, and BGE-Reranker-v2-Gemma. DeepSeek-R1-Distill-Qwen-7B is additionally evaluated in embedding, reranking, and generative-selection configurations. The bi-encoder separately encodes the context and candidate; the cross-encoder jointly processes one context–candidate pair. Listwise supervision does not require jointly encoding the full candidate list at inference. Table 6 lists the model release dates recorded in this draft.

A.4 Training Configurations and Optimization

Full SFT updates model parameters using task-specific supervision. The embedding objective is contrastive; the reranker experiments compare pointwise and listwise supervision as described in the methodology. Training-loss curves summarize optimization behavior, but do not by themselves establish generalization. Figure 5 shows the training loss curves of the eight primary retrieval models.

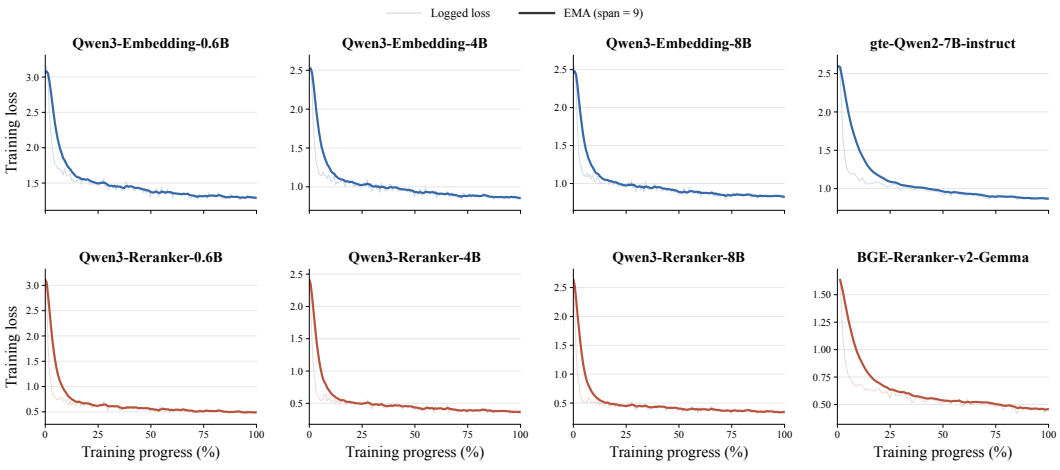


Figure 5: Training loss curves of the eight primary retrieval models under full listwise fine-tuning.

Table 5: Training and evaluation protocols. Historical runs are reported separately because their training configurations are not fully matched.

Item	Configuration / reporting convention
Model and data	Qwen3-Embedding-0.6B; 195,301 original training rows and 3,132 test queries. Mechanism analyses use the historical FT checkpoint `checkpoint-1526`.
Training recipes	
Large deletion runs	R6, R7, R6q, and T3: 4 GPUs $\times$ 4 rows/GPU, gradient accumulation 8, effective batch size 128; 16 contrastive groups per loss computation; ZeRO-3. One epoch; learning rate 10^{-5}, cosine schedule, warmup ratio 0.05, weight decay 0.01, gradient clipping 0.5; maximum training length 32,768 tokens; training seed 42.
Single-GPU small runs	1 GPU $\times$ 8 rows/GPU, gradient accumulation 4, effective batch size 32; 8 contrastive groups per loss computation. One epoch; learning rate 10^{-5}, constant schedule, no warmup, weight decay 0.01, gradient clipping 1.0; maximum training length 32,768 tokens.
Historical full-data R0	Gradient accumulation 16, versus 8 in the current deletion runs. Historical cross-GPU loss behavior has not been fully verified as equivalent; R0 is a descriptive reference, not a recipe-matched control.
Historical small runs	Four-GPU small runs: 4×2 rows, accumulation 4, effective batch 32, DDP. Historical random-2k: 4×1 rows, accumulation 8, effective batch 32, ZeRO-3. These are separate from the single-GPU runs.
Historical Figs. 3 and 5	The supplied archive does not establish a complete figure-to-run mapping. The historical recipes above should not be assigned to either figure without checking the original run configuration.
Loss and negatives	InfoNCE; retain the original 10–14 negatives per row. Gradient accumulation does not enlarge the contrastive group in a single loss computation. The code-default contrastive temperature is 0.1; historical overrides are not exhaustively verified.
Hardware and seeds	Main-table result: H200; historical mechanism baseline: H800. Three training seeds (42, 43, 44) are verified for Rand1, S_{base1}, and S_{FT1} under the single-GPU recipe; additional selection probes use seed 42. Sampling/grouping seed: 20260925.
Evaluation protocols and denominators	
Shared evaluation	FP32, SDPA, maximum input length 8,192 tokens, evaluation batch size 16; tie-breaking by `reverse_candidate_index`. Documents receive no instruction.
Original	No query instruction; rank the original per-query candidate set (11–15 candidates after deduplication). Top-1 accuracy is computed over 3,132 queries.
All-Positive	Queries use the ABC instruction below; each query is ranked against the same pool of 3,035 positive documents. Top-1 accuracy is computed over 3,132 queries. The $3{,}132 \times 3{,}035 = 9{,}505{,}620$ query–document scores are not the accuracy denominator.
Recovery / regression	Freeze the Original Base-fail set $\mathcal{F}$ ($\lvert\mathcal{F}\rvert = 2{,}033$) and Base-correct set $\mathcal{C}$ ($\lvert\mathcal{C}\rvert = 1{,}099$). For model M, recovery is $\lvert\{q \in \mathcal{F} : M \text{ correct}\}\rvert/2{,}033$; regression is $\lvert\{q \in \mathcal{C} : M \text{ incorrect}\}\rvert/1{,}099$. Subset analyses use and disclose their corresponding subset denominators.
Uncertainty	Three-seed summaries report mean $\pm$ sample SD. Wilson intervals and query-level bootstrap intervals quantify query-level uncertainty, not variation across training seeds.
77.39% vs. 76.21%	The main table reports 77.39% Original Top-1 from the H200 run. Mechanism analyses use the historical H800 run with 76.21% Original Top-1 and its per-query predictions. These results are not interchangeable. Their 1.18-percentage-point difference does not identify a hardware effect or estimate seed variance.

ABC query format: `Instruct: Given a web search query, retrieve relevant passages that answer the query`
`Query:` *[query text]*. The newline before `Query:` is part of the format. The cached query embeddings used for geometric analyses also use this instruction, while Original recovery labels come from evaluation without query instruction.

Table 6: Model release dates.

Models	Date
GritLM-7B	2024.02.11
BGE-Reranker-v2-Gemma	2024.03.16
gte-Qwen2-7B-instruct	2024.06.16
bge-en-icl	2024.07.25
stella_en_400M_v5	2024.07.31
stella_en_1.5B_v5	2024.07.31
DeepSeek-R1-Distill-Qwen-7B	2025.01.20
ReasonIR-8B	2025.04.03
Qwen3-Reranker-0.6B	2025.05.29
Qwen3-Reranker-8B	2025.05.29
Qwen3-Embedding-0.6B	2025.06.03
Qwen3-Embedding-4B	2025.06.03
Qwen3-Embedding-8B	2025.06.03
Qwen3-Reranker-4B	2025.06.03

A.5 Evaluation Metrics and Reporting Conventions

For a query whose annotated inspiration has rank r, top-k accuracy records whether $r \leq k$. Mean positive rank averages r, and MRR averages $1/r$ across queries. Top-k and pairwise accuracies are reported as percentages. MRR is dimensionless.

B Full Retrieval Results and Candidate-Set Analysis

B.1 Detailed Results on the Original Candidate Set

Table 7 reports the complete Original-set results. In the following tables, Qwen-E and Qwen-R denote Qwen3 embedding and reranking models; DS-E and DS-R denote the corresponding DeepSeek-R1-Distill-Qwen models. GTE-7B and BGE-Gemma abbreviate GTE-Qwen2-7B-Instruct and BGE-Reranker-v2-Gemma. The suffixes list and point denote listwise and pointwise full fine-tuning. Rank denotes mean positive rank (lower is better).

Table 7: Results on the Original test set. -FT denotes full fine-tuning; Objective specifies the supervision used. Top-k and pairwise accuracies are percentages.

Models	Objective	Top-1 Acc.	Top-2 Acc.	Top-3 Acc.	Top-4 Acc.	Top-5 Acc.	Top-10 Acc.	Mean Positive Rank↓	MRR	Pairwise Acc. (%)
DeepSeek-R1-Distill-Qwen-7B-Embedding	–	8.68	16.19	23.56	30.84	38.38	70.98	7.49	0.25	53.48
DeepSeek-R1-Distill-Qwen-7B-Embedding-FT	Listwise	71.58	88.15	94.35	97.29	98.69	99.81	1.52	0.83	96.29
DeepSeek-R1-Distill-Qwen-7B-Reranker	–	10.60	18.97	26.05	33.46	40.20	70.53	7.35	0.27	54.50
DeepSeek-R1-Distill-Qwen-7B-Reranker-FT	Listwise	83.52	94.48	97.19	98.34	99.36	99.97	1.28	0.90	98.00
Qwen3-Embedding-0.6B	–	35.09	55.84	70.11	79.82	86.14	97.83	3.01	0.56	85.59
Qwen3-Embedding-0.6B-FT	Listwise	77.39	91.38	95.50	97.89	98.85	99.97	1.40	0.87	97.10
Qwen3-Embedding-4B	–	37.61	58.65	73.08	82.95	90.07	98.79	2.76	0.58	87.42
Qwen3-Embedding-4B-FT	Listwise	82.12	93.90	97.57	98.56	99.52	100.00	1.29	0.90	97.94
Qwen3-Embedding-8B	–	38.38	60.19	74.58	83.40	90.55	98.60	2.71	0.59	87.75
Qwen3-Embedding-8B-FT	Listwise	82.50	93.71	97.35	98.69	99.36	100.00	1.29	0.90	97.90
gte-Qwen2-7B-instruct	–	38.28	60.09	74.36	83.84	89.78	98.44	2.73	0.59	87.61
gte-Qwen2-7B-instruct-FT	Listwise	74.43	89.02	93.84	95.95	97.64	99.23	1.56	0.84	95.95
Qwen3-Reranker-0.6B	–	43.14	64.97	78.83	87.39	91.95	98.63	2.51	0.63	89.26
Qwen3-Reranker-0.6B-FT	Pointwise	79.73	92.31	96.20	97.99	98.88	100.00	1.36	0.88	97.47
Qwen3-Reranker-0.6B-FT	Listwise	81.13	93.13	96.87	98.31	99.07	100.00	1.32	0.89	97.71
Qwen3-Reranker-4B	–	52.71	74.71	87.01	92.69	95.95	99.62	2.03	0.70	92.69
Qwen3-Reranker-4B-FT	Pointwise	84.29	94.70	97.51	98.79	99.33	99.94	1.26	0.91	98.15
Qwen3-Reranker-4B-FT	Listwise	85.63	95.24	97.83	98.82	99.36	100.00	1.24	0.92	98.34
Qwen3-Reranker-8B	–	52.97	74.94	87.13	92.46	95.66	99.30	2.05	0.71	92.50
Qwen3-Reranker-8B-FT	Pointwise	85.57	95.02	97.45	98.60	99.30	100.00	1.25	0.92	98.27
Qwen3-Reranker-8B-FT	Listwise	86.33	95.82	98.02	99.07	99.55	100.00	1.22	0.92	98.49
BGE-Reranker-v2-Gemma	–	42.43	64.72	78.45	86.75	91.19	98.63	2.55	0.62	89.01
BGE-Reranker-v2-Gemma-FT	Pointwise	81.80	93.36	96.84	98.50	99.17	99.87	1.32	0.89	97.74
BGE-Reranker-v2-Gemma-FT	Listwise	82.38	94.13	97.13	98.34	99.17	99.90	1.30	0.90	97.85
ReasonIR-8B	–	34.10	54.44	68.20	78.10	86.33	98.15	3.02	0.55	85.53
bge-en-icl	–	35.82	55.87	69.57	78.16	84.61	96.42	3.17	0.56	84.47
stella_en_400M_v5	–	38.41	60.79	75.45	83.81	89.75	98.28	2.75	0.59	87.49
stella_en_1.5B_v5	–	40.61	62.87	76.79	86.14	91.06	98.37	2.62	0.61	88.37
GritLM-7B	–	39.30	61.14	75.80	84.23	90.20	98.85	2.66	0.60	88.12

B.2 Results under Controlled Negative Distributions

Tables 8, 9, and 10 provide the complete results for the three controlled candidate distributions.

Table 8: Results on the All Hard test set. -FT denotes full fine-tuning; Objective specifies the supervision used. Top-k and pairwise accuracies are percentages.

Models	Objective	Top-1 Acc.	Top-2 Acc.	Top-3 Acc.	Top-4 Acc.	Top-5 Acc.	Top-10 Acc.	Mean Positive Rank↓	MRR	Pairwise Acc. (%)
DeepSeek-R1-Distill-Qwen-7B-Embedding	–	8.75	16.32	23.12	29.85	36.14	68.58	7.70	0.24	52.14
DeepSeek-R1-Distill-Qwen-7B-Embedding-FT	Listwise	73.18	85.31	90.77	93.65	95.75	99.23	1.72	0.83	94.85
DeepSeek-R1-Distill-Qwen-7B-Reranker	–	9.10	17.50	25.35	32.31	38.63	69.16	7.54	0.25	53.28
DeepSeek-R1-Distill-Qwen-7B-Reranker-FT	Listwise	85.22	93.52	96.17	97.64	98.47	99.81	1.32	0.91	97.69
Qwen3-Embedding-0.6B	–	37.77	54.25	64.85	71.74	77.97	93.97	3.61	0.55	81.38
Qwen3-Embedding-0.6B-FT	Listwise	79.12	89.78	94.00	96.23	97.48	99.71	1.49	0.87	96.51
Qwen3-Embedding-4B	–	43.20	59.39	70.63	78.13	83.27	96.49	3.09	0.60	85.07
Qwen3-Embedding-4B-FT	Listwise	84.64	93.61	96.42	97.64	98.44	99.81	1.32	0.91	97.71
Qwen3-Embedding-8B	–	45.08	62.26	72.73	80.01	84.96	96.42	2.97	0.62	85.96
Qwen3-Embedding-8B-FT	Listwise	85.28	93.30	96.23	97.61	98.28	99.84	1.32	0.91	97.69
gte-Qwen2-7B-instruct	–	45.69	62.55	73.12	79.69	84.96	96.68	2.92	0.62	86.26
gte-Qwen2-7B-instruct-FT	Listwise	77.04	88.70	92.91	95.02	96.42	99.11	1.60	0.86	95.69
Qwen3-Reranker-0.6B	–	45.21	61.65	71.55	78.10	82.63	95.56	3.10	0.61	85.13
Qwen3-Reranker-0.6B-FT	Pointwise	82.69	91.67	94.76	96.42	97.41	99.58	1.43	0.89	97.02
Qwen3-Reranker-0.6B-FT	Listwise	83.62	92.34	95.69	97.22	98.08	99.74	1.37	0.90	97.42
Qwen3-Reranker-4B	–	55.36	73.15	81.13	86.30	90.39	98.56	2.35	0.70	90.43
Qwen3-Reranker-4B-FT	Pointwise	86.91	93.58	96.26	97.54	98.31	99.62	1.31	0.92	97.80
Qwen3-Reranker-4B-FT	Listwise	87.80	94.80	96.78	97.99	98.63	99.84	1.27	0.93	98.14
Qwen3-Reranker-8B	–	56.10	72.51	81.39	87.16	90.52	97.99	2.37	0.71	90.26
Qwen3-Reranker-8B-FT	Pointwise	87.80	94.28	96.81	97.80	98.44	99.62	1.29	0.92	97.96
Qwen3-Reranker-8B-FT	Listwise	88.31	95.15	97.29	98.37	98.75	99.74	1.25	0.93	98.28
BGE-Reranker-v2-Gemma	–	44.28	61.97	71.68	78.48	83.65	95.75	3.07	0.61	85.38
BGE-Reranker-v2-Gemma-FT	Pointwise	83.81	92.24	95.40	96.93	97.96	99.68	1.38	0.90	97.30
BGE-Reranker-v2-Gemma-FT	Listwise	84.74	93.77	96.55	97.67	98.15	99.65	1.33	0.91	97.62
ReasonIR-8B	–	33.65	46.78	55.87	63.31	69.80	89.24	4.36	0.50	75.98
bge-en-icl	–	36.69	53.00	62.07	69.22	75.48	92.98	3.78	0.54	80.17
stella_en_400M_v5	–	39.43	56.61	67.53	75.19	81.55	96.17	3.27	0.57	83.77
stella_en_1.5B_v5	–	41.95	59.00	69.86	77.39	82.31	96.39	3.14	0.59	84.71
GritLM-7B	–	38.60	54.34	64.81	72.83	78.35	93.65	3.60	0.56	81.43

Table 9: Results on the All Random test set. -FT denotes full fine-tuning; Objective specifies the supervision used. Top-k and pairwise accuracies are percentages.

Models	Objective	Top-1 Acc.	Top-2 Acc.	Top-3 Acc.	Top-4 Acc.	Top-5 Acc.	Top-10 Acc.	Mean Positive Rank↓	MRR	Pairwise Acc. (%)
DeepSeek-R1-Distill-Qwen-7B-Embedding	–	10.50	18.61	25.51	32.22	39.14	69.92	7.43	0.26	54.08
DeepSeek-R1-Distill-Qwen-7B-Embedding-FT	Listwise	89.56	95.85	97.99	98.91	99.39	99.87	1.20	0.94	98.59
DeepSeek-R1-Distill-Qwen-7B-Reranker	–	12.45	21.68	29.21	35.66	42.46	70.91	7.21	0.29	55.67
DeepSeek-R1-Distill-Qwen-7B-Reranker-FT	Listwise	89.21	96.71	98.75	99.36	99.65	100.00	1.17	0.94	98.80
Qwen3-Embedding-0.6B	–	73.15	82.60	88.15	90.90	93.07	98.02	1.93	0.82	93.36
Qwen3-Embedding-0.6B-FT	Listwise	88.31	95.95	97.96	98.85	99.36	99.97	1.21	0.93	98.52
Qwen3-Embedding-4B	–	76.79	87.10	91.19	93.52	95.27	98.60	1.70	0.85	94.97
Qwen3-Embedding-4B-FT	Listwise	90.20	96.49	98.50	99.20	99.58	99.97	1.17	0.94	98.81
Qwen3-Embedding-8B	–	79.50	88.12	92.43	94.60	95.91	98.75	1.62	0.87	95.58
Qwen3-Embedding-8B-FT	Listwise	90.49	96.68	98.34	99.07	99.43	99.97	1.17	0.94	98.80
gte-Qwen2-7B-instruct	–	73.56	84.93	89.88	92.31	94.48	98.37	1.82	0.83	94.13
gte-Qwen2-7B-instruct-FT	Listwise	84.55	93.17	96.01	97.41	98.05	99.07	1.39	0.90	97.21
Qwen3-Reranker-0.6B	–	79.28	88.63	92.31	94.13	95.43	98.63	1.65	0.86	95.40
Qwen3-Reranker-0.6B-FT	Pointwise	85.60	94.89	97.67	99.07	99.39	100.00	1.24	0.92	98.31
Qwen3-Reranker-0.6B-FT	Listwise	87.20	95.66	98.15	99.07	99.52	100.00	1.21	0.93	98.52
Qwen3-Reranker-4B	–	85.86	92.91	95.69	96.84	97.67	99.46	1.37	0.91	97.37
Qwen3-Reranker-4B-FT	Pointwise	89.50	97.16	98.72	99.46	99.65	100.00	1.16	0.94	98.88
Qwen3-Reranker-4B-FT	Listwise	91.16	97.25	98.95	99.52	99.74	100.00	1.14	0.95	99.04
Qwen3-Reranker-8B	–	84.42	91.79	94.54	96.10	97.19	99.39	1.43	0.90	96.93
Qwen3-Reranker-8B-FT	Pointwise	90.55	97.22	98.91	99.62	99.68	100.00	1.14	0.95	98.99
Qwen3-Reranker-8B-FT	Listwise	91.41	97.80	99.11	99.58	99.87	100.00	1.12	0.95	99.14
BGE-Reranker-v2-Gemma	–	76.60	86.02	90.84	93.58	95.24	98.85	1.72	0.85	94.89
BGE-Reranker-v2-Gemma-FT	Pointwise	88.15	96.14	98.34	99.17	99.46	99.97	1.19	0.93	98.62
BGE-Reranker-v2-Gemma-FT	Listwise	88.41	96.68	98.44	99.17	99.65	99.97	1.18	0.93	98.71
ReasonIR-8B	–	75.73	86.24	90.17	92.94	94.54	98.79	1.76	0.84	94.59
bge-en-icl	–	67.24	77.59	83.11	86.08	88.47	96.30	2.35	0.77	90.34
stella_en_400M_v5	–	72.13	83.11	88.19	90.87	92.85	97.77	1.96	0.81	93.15
stella_en_1.5B_v5	–	74.33	84.90	89.18	91.95	93.36	97.96	1.87	0.83	93.80
GritLM-7B	–	77.11	86.72	91.51	93.81	95.47	98.85	1.68	0.85	95.14

Table 10: Results on the Decoy Cluster test set. -FT denotes full fine-tuning; Objective specifies the supervision used. Top-k and pairwise accuracies are percentages.

Models	Objective	Top-1 Acc.	Top-2 Acc.	Top-3 Acc.	Top-4 Acc.	Top-5 Acc.	Top-10 Acc.	Mean Positive Rank↓	MRR	Pairwise Acc. (%)
DeepSeek-R1-Distill-Qwen-7B-Embedding	–	10.03	18.07	26.09	33.01	39.30	70.24	7.40	0.26	54.29
DeepSeek-R1-Distill-Qwen-7B-Embedding-FT	Listwise	91.09	96.42	98.21	98.72	99.11	99.87	1.18	0.95	98.70
DeepSeek-R1-Distill-Qwen-7B-Reranker	–	11.02	20.31	27.62	34.36	40.96	69.51	7.41	0.27	54.21
DeepSeek-R1-Distill-Qwen-7B-Reranker-FT	Listwise	91.19	97.41	99.04	99.49	99.68	100.00	1.14	0.95	99.02
Qwen3-Embedding-0.6B	–	73.85	83.49	88.47	91.28	92.91	97.86	1.92	0.82	93.44
Qwen3-Embedding-0.6B-FT	Listwise	90.77	96.52	98.50	99.23	99.65	99.97	1.16	0.95	98.87
Qwen3-Embedding-4B	–	78.77	86.85	91.19	93.33	94.92	98.91	1.69	0.86	95.06
Qwen3-Embedding-4B-FT	Listwise	92.59	97.54	98.72	99.46	99.74	99.94	1.13	0.96	99.10
Qwen3-Embedding-8B	–	80.75	89.37	92.46	94.51	96.01	98.72	1.60	0.87	95.70
Qwen3-Embedding-8B-FT	Listwise	93.23	97.38	98.95	99.39	99.68	99.94	1.12	0.96	99.14
gte-Qwen2-7B-instruct	–	75.99	86.37	90.20	93.07	94.86	98.34	1.76	0.84	94.58
gte-Qwen2-7B-instruct-FT	Listwise	88.03	94.54	96.84	97.77	98.31	99.43	1.30	0.93	97.84
Qwen3-Reranker-0.6B	–	80.59	88.63	91.95	94.06	95.11	98.53	1.65	0.87	95.41
Qwen3-Reranker-0.6B-FT	Pointwise	88.22	95.82	98.44	99.20	99.49	99.94	1.20	0.93	98.63
Qwen3-Reranker-0.6B-FT	Listwise	89.97	96.68	98.50	99.17	99.58	100.00	1.17	0.94	98.83
Qwen3-Reranker-4B	–	86.46	93.14	95.53	97.03	97.80	99.39	1.36	0.91	97.44
Qwen3-Reranker-4B-FT	Pointwise	91.63	97.67	99.04	99.52	99.65	99.97	1.13	0.95	99.07
Qwen3-Reranker-4B-FT	Listwise	93.01	98.15	99.23	99.58	99.78	100.00	1.11	0.96	99.25
Qwen3-Reranker-8B	–	85.44	92.66	95.21	96.58	97.16	99.20	1.41	0.91	97.06
Qwen3-Reranker-8B-FT	Pointwise	91.89	97.86	99.20	99.55	99.74	100.00	1.12	0.95	99.15
Qwen3-Reranker-8B-FT	Listwise	93.52	98.28	99.43	99.68	99.87	100.00	1.09	0.96	99.36
BGE-Reranker-v2-Gemma	–	77.04	86.11	90.74	93.23	94.48	98.69	1.73	0.85	94.80
BGE-Reranker-v2-Gemma-FT	Pointwise	90.45	97.09	98.82	99.33	99.58	99.97	1.15	0.95	98.91
BGE-Reranker-v2-Gemma-FT	Listwise	90.71	97.09	98.79	99.39	99.71	99.97	1.15	0.95	98.92
ReasonIR-8B	–	76.02	85.28	89.62	92.05	93.65	98.28	1.83	0.84	94.09
bge-en-icl	–	69.25	78.96	83.30	86.81	88.83	96.17	2.31	0.78	90.65
stella_en_400M_v5	–	74.74	84.55	88.79	91.38	92.98	97.99	1.89	0.83	93.61
stella_en_1.5B_v5	–	75.96	85.63	89.66	92.15	93.52	98.08	1.83	0.84	94.08
GritLM-7B	–	78.86	87.04	90.71	93.23	94.80	98.72	1.70	0.86	95.03

Table 11: Results on the Original test set without background survey during training and inference. MRR: Mean Reciprocal Rank, FT: Fine-Tuned.

Models	Background	Top-1 Acc.	Top-2 Acc.	Top-3 Acc.	Top-4 Acc.	Top-5 Acc.	Top-10 Acc.	MRR
Qwen3-Embedding-0.6B-FT	w/o	71.14	87.58	94.09	96.58	98.21	99.84	0.83
Qwen3-Embedding-0.6B-FT	w/	77.39	91.38	95.50	97.89	98.85	99.97	0.87
Qwen3-Embedding-4B-FT	w/o	77.11	91.25	95.98	97.99	98.82	99.97	0.87
Qwen3-Embedding-4B-FT	w/	82.12	93.90	97.57	98.56	99.52	100.00	0.90
Qwen3-Embedding-8B-FT	w/o	78.13	91.63	96.14	98.02	98.72	99.97	0.87
Qwen3-Embedding-8B-FT	w/	82.50	93.71	97.35	98.69	99.36	100.00	0.90
gte-Qwen2-7B-instruct-FT	w/o	68.93	85.95	91.73	94.76	96.71	99.23	0.81
gte-Qwen2-7B-instruct-FT	w/	74.43	89.02	93.84	95.95	97.64	99.23	0.84
Qwen3-Reranker-0.6B-FT	w/o	78.93	92.05	96.58	97.99	99.01	100.00	0.88
Qwen3-Reranker-0.6B-FT	w/	81.13	93.13	96.87	98.31	99.07	100.00	0.89
Qwen3-Reranker-4B-FT	w/o	84.55	94.96	97.70	98.79	99.43	100.00	0.91
Qwen3-Reranker-4B-FT	w/	85.63	95.24	97.83	98.82	99.36	100.00	0.92
Qwen3-Reranker-8B-FT	w/o	85.25	95.31	97.83	98.72	99.33	100.00	0.92
Qwen3-Reranker-8B-FT	w/	86.33	95.82	98.02	99.07	99.55	100.00	0.92
BGE-Reranker-v2-Gemma-FT	w/o	80.08	92.62	96.52	98.12	98.98	99.94	0.88
BGE-Reranker-v2-Gemma-FT	w/	82.38	94.13	97.13	98.34	99.17	99.90	0.90

B.3 Ranking Performance on the All-Positive Pool

Table 1 reports the mean rank of the annotated inspiration in the larger All-Positive pool. This evaluates competition against inspirations associated with other problems, but does not constitute exhaustive relevance annotation or retrieval over the full scientific literature.

B.4 Matched-Backbone Comparison with MS-IR

Table 12 compares retrieval formulations built on DeepSeek-R1-Distill-Qwen-7B and distinguishes the 150,218-example runs from rows marked Full. A common backbone controls model family and nominal parameter scale, but does not imply identical objectives or compute budgets. Table 13 separately compares base and fine-tuned models on 2,377 queries. Missing results are not inferred from other runs.

Table 12: Retrieval performance using the DeepSeek-R1-Distill-Qwen-7B backbone. MS-IR uses full SFT; Embedding and Reranker use listwise full SFT. Full denotes rows without the #150218 suffix in the source. The first four metrics are Top-1 accuracy (%); Proposed Rank reports mean positive rank in the 3,035-paper candidate pool.

Models	Training Data	Original	All Random	All Hard	Decoy Cluster	Proposed Rank↓
MS-IR-7B	Full	54.37	78.58	53.42	77.65	742.50
DeepSeek-R1-Distill-Qwen-7B-Embedding-FT	Full	71.58	89.56	73.18	91.09	45.42
DeepSeek-R1-Distill-Qwen-7B-Embedding-FT	150,218	70.50	88.89	72.48	90.90	47.50
DeepSeek-R1-Distill-Qwen-7B-Reranker-FT	Full	83.52	89.21	85.22	91.19	41.89
DeepSeek-R1-Distill-Qwen-7B-Reranker-FT	150,218	81.61	89.66	83.78	91.95	41.43

Table 13: Original-setting results on 2,377 test queries. Both Embedding and Reranker use DeepSeek-R1-Distill-Qwen-7B. Base denotes the model before task-specific fine-tuning; SFT denotes listwise full fine-tuning on 150,218 examples. Top-k and pairwise accuracies are percentages. Rank denotes mean positive rank.

Models	Setting	Top-1 Acc.	Top-2 Acc.	Top-3 Acc.	Top-4 Acc.	Top-5 Acc.	Top-10 Acc.	Mean Positive Rank↓	MRR	Pairwise Acc. (%)
DeepSeek-R1-Distill-Qwen-7B-Embedding	Base	9.05	16.87	24.57	32.06	39.25	72.23	7.36	0.25	54.59
DeepSeek-R1-Distill-Qwen-7B-Embedding-FT	SFT	78.84	92.76	97.18	98.53	99.45	99.92	1.34	0.88	97.57
DeepSeek-R1-Distill-Qwen-7B-Reranker	Base	11.91	20.49	27.89	35.46	42.11	72.02	7.15	0.28	56.08
DeepSeek-R1-Distill-Qwen-7B-Reranker-FT	SFT	86.33	96.30	98.53	99.37	99.58	99.96	1.20	0.92	98.55

B.5 Retrieval without Background Information

Table 11 removes background during both training and inference. All eight fine-tuned configurations have lower Original-set Top-1 accuracy without background. This supports its usefulness in the evaluated setup, but does not isolate the effect of removing background only at inference.

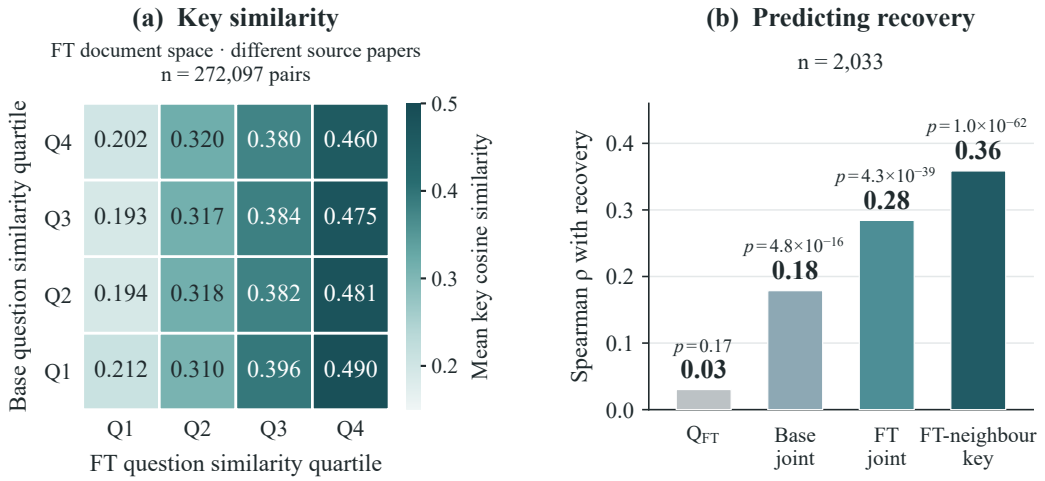


Figure 6: The learned question similarity is key-predictive beyond topic. (a) For $\tilde{2}72$k (test, training) question pairs with different source papers, mean cosine between the test key and the training key, binned by question similarity under the base (rows) and fine-tuned (columns) encoders: key similarity rises along the fine-tuned axis (anti-diagonal difference 0.289 [0.278, 0.299]) and is flat along the base axis (0.074 [0.068, 0.080]); partial Spearman 0.43 controlling base similarity. (b) Spearman correlation with recovery (n = 2,033): the nearest training question alone predicts nothing (r = 0.03, p = 0.17); the nearest training pair predicts weakly under the base encoder (0.18) and more strongly under the fine-tuned encoder (0.28); the similarity between the keys of the ten nearest questions and the needed key predicts best (0.36). Fine-tuned-space measures partly reflect the model's own geometry; causal evidence is in Fig. M1.

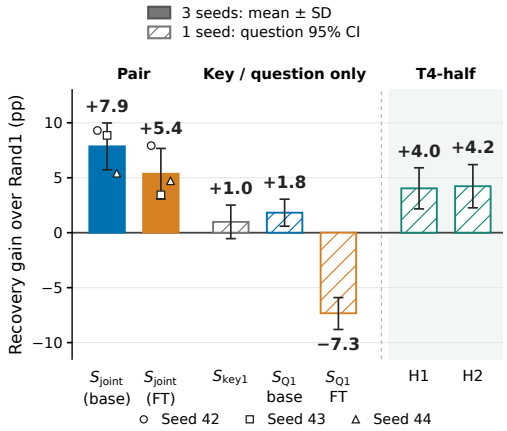
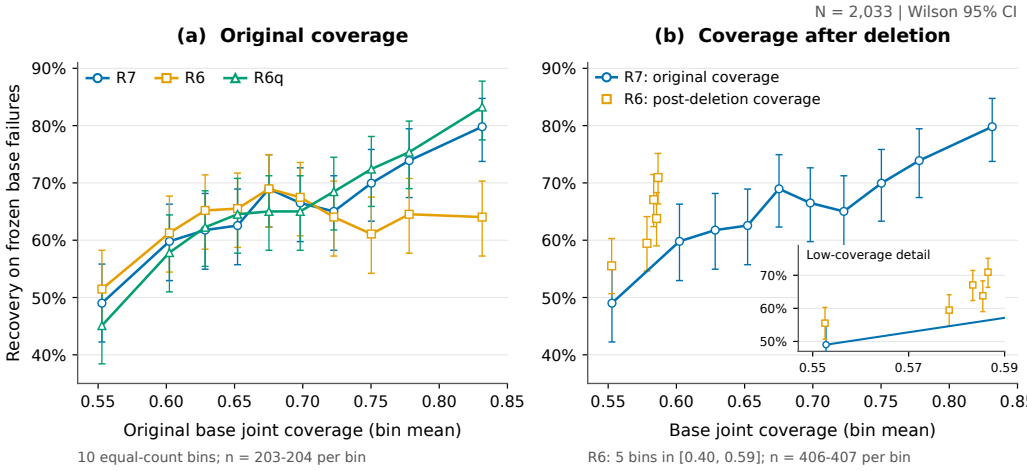


Figure 7: Recovery gain of 2k-row training sets over 2k random rows (three seeds), by selection rule: rows nearest to each test pair in problem and inspiration (base or fine-tuned space), in inspiration only, or in problem only; T4-half selects for a random half of the test problems and reports both halves. Selection uses the annotated inspiration and probes the mechanism only. Recipe in Table 5.



Paired question bootstrap, 95% CI

Comparison	R7 − R6	R6q − R6	R7 − R6q	T3 − R7
Difference (pp)	2.36	2.56	−0.20	−1.52
95% CI	[0.84, 3.89]	[0.84, 4.23]	[−1.52, 1.13]	[−2.80, −0.30]

Figure 8: Recovery by coverage bin after deleting 34% of the training rows: at random (R7), those resembling a test pair (R6), those resembling a test problem only (R6q), or each test problem's 30 nearest rows (T3). (a) Only the highest-coverage bins lose recovery, and only under R6. (b) Problems R6 de-covered still recover more than naturally low-coverage ones. Definitions in Table 5.

C Training-Scale and Supervision Ablations

C.1 Training Subset Construction and Budget Control

The scaling experiments vary training-set size and evaluate the resulting models on the retrieval task. These are data-scale comparisons, not evidence of equal total optimization compute across sizes. The separate 2,377-query experiment is not pooled with the 3,132-query main evaluation.

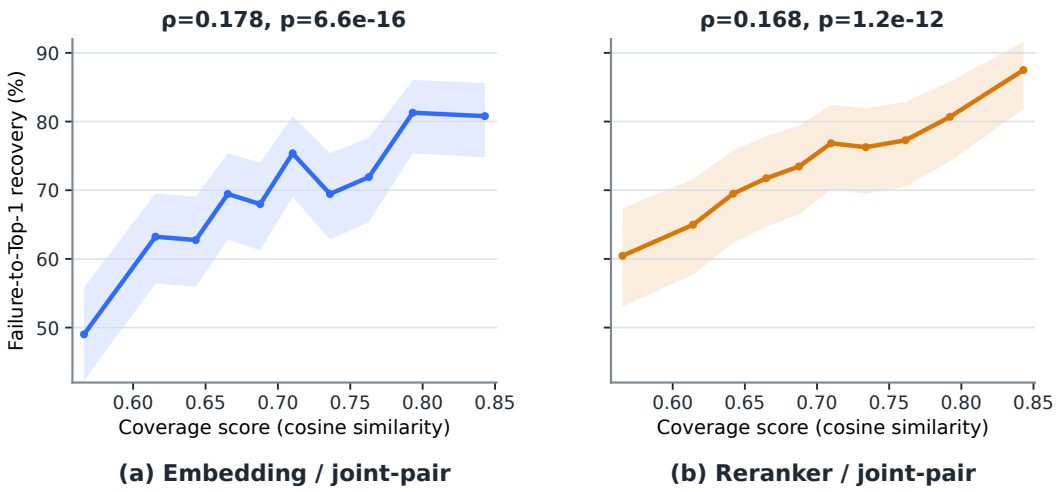


Figure 9: Trainability is associated with problem–inspiration coverage.

C.2 Complete Learning Curves across Data Scales

Figure 4 shows larger gains at small training-data scales and smaller gains toward the largest evaluated settings. This is an empirical trend over the measured range, not an extrapolated scaling law or evidence that further data cannot help.

C.3 Pointwise versus Listwise Supervision

Table 17 compares the two objectives for reranking. Listwise training improves Original-set Top-1 accuracy for all four configurations by 0.58–1.40 percentage points. These results support the selected objective within the evaluated settings, rather than superiority across arbitrary tasks or negative-sampling schemes.

C.4 Effects of Negative Count and In-Batch Negatives

Increasing the number of negatives from 7 to 14 improves Top-1 accuracy for every model in Table 14, with gains ranging from 0.07 to 1.88 points.

Removing in-batch negatives has mixed effects in Table 15: it improves some configurations and reduces accuracy in others.

Table 14: Ablation on the number of negatives.

Models	#neg.	Top-1 Acc.	Top-2 Acc.	Top-3 Acc.	Top-4 Acc.	Top-5 Acc.	Top-10 Acc.	Mean Positive Rank↓	MRR	Pairwise Acc. (%)
Qwen3-Embedding-0.6B-FT	7	75.51	90.93	96.10	98.05	99.27	99.94	1.41	0.86	97.04
Qwen3-Embedding-0.6B-FT	14	77.39	91.38	95.50	97.89	98.85	99.97	1.40	0.87	97.10
Qwen3-Embedding-4B-FT	7	81.19	93.68	97.32	98.82	99.43	99.97	1.30	0.89	97.86
Qwen3-Embedding-4B-FT	14	82.12	93.90	97.57	98.56	99.52	100.00	1.29	0.90	97.94
Qwen3-Embedding-8B-FT	7	82.22	93.68	97.35	98.69	99.52	100.00	1.29	0.90	97.92
Qwen3-Embedding-8B-FT	14	82.50	93.71	97.35	98.69	99.36	100.00	1.29	0.90	97.90
gte-Qwen2-7B-instruct-FT	7	74.36	88.25	93.71	95.95	97.51	99.30	1.58	0.84	95.87
gte-Qwen2-7B-instruct-FT	14	74.43	89.02	93.84	95.95	97.64	99.23	1.56	0.84	95.95
Qwen3-Reranker-0.6B-FT	7	79.98	92.75	96.36	97.99	98.95	100.00	1.35	0.88	97.54
Qwen3-Reranker-0.6B-FT	14	81.13	93.13	96.87	98.31	99.07	100.00	1.32	0.89	97.71
Qwen3-Reranker-4B-FT	7	84.67	94.83	97.77	98.75	99.36	100.00	1.25	0.91	98.24
Qwen3-Reranker-4B-FT	14	85.63	95.24	97.83	98.82	99.36	100.00	1.24	0.92	98.34
Qwen3-Reranker-8B-FT	7	84.87	95.59	97.80	98.85	99.39	100.00	1.24	0.91	98.34
Qwen3-Reranker-8B-FT	14	86.33	95.82	98.02	99.07	99.55	100.00	1.22	0.92	98.49
BGE-Reranker-v2-Gemma-FT	7	81.29	93.20	96.90	98.47	98.95	99.87	1.33	0.89	97.63
BGE-Reranker-v2-Gemma-FT	14	82.38	94.13	97.13	98.34	99.17	99.90	1.30	0.90	97.85

Table 15: Results with and without in-batch negatives on the Original test set. All models use listwise full fine-tuning; w/ is the default setting.

Models	In-Batch Negatives	Top-1 Acc.	Top-2 Acc.	Top-3 Acc.	Top-4 Acc.	Top-5 Acc.	Top-10 Acc.	Mean Positive Rank↓	MRR	Pairwise Acc. (%)
Qwen3-Embedding-0.6B-FT	w/o	77.71	91.38	95.63	97.64	98.72	99.94	1.41	0.87	97.07
Qwen3-Embedding-0.6B-FT	w/	77.39	91.38	95.50	97.89	98.85	99.97	1.40	0.87	97.10
Qwen3-Embedding-4B-FT	w/o	81.48	93.45	96.93	98.34	99.14	99.87	1.32	0.89	97.70
Qwen3-Embedding-4B-FT	w/	82.12	93.90	97.57	98.56	99.52	100.00	1.29	0.90	97.94
Qwen3-Embedding-8B-FT	w/o	81.77	93.68	96.78	98.24	98.95	99.87	1.32	0.89	97.69
Qwen3-Embedding-8B-FT	w/	82.50	93.71	97.35	98.69	99.36	100.00	1.29	0.90	97.90
gte-Qwen2-7B-instruct-FT	w/o	76.44	89.43	93.58	95.66	97.09	99.43	1.55	0.85	96.04
gte-Qwen2-7B-instruct-FT	w/	74.43	89.02	93.84	95.95	97.64	99.23	1.56	0.84	95.95

C.5 Instruction Templates and Train–Test Sensitivity

Instruction effects are less uniform (Table 16). For the pretrained Qwen3-Embedding-8B model, adding an instruction changes Top-1 accuracy by up to 4.88 points, whereas its nine train–test instruction combinations after full fine-tuning remain within a 1.08-point range. MIR-specific instructions provide a small gain for Qwen3-Reranker-8B (86.91% versus 86.33% with matched default instructions), but reduce performance for BGE-Reranker-v2-Gemma (80.24% versus 82.38%). Thus, task-specific fine-tuning substantially reduces sensitivity to instruction wording, while specialized instructions do not offer a uniform advantage. These ablations show architecture-dependent effects and do not constitute a compute-controlled comparison of all training choices.

Table 16: Effect of training and inference instructions. -FT denotes listwise full fine-tuning; – indicates that no task-specific fine-tuning was applied.

Models	Training Instruction	Inference Instruction	Top-1 Acc.	Top-2 Acc.	Top-3 Acc.	Top-4 Acc.	Top-5 Acc.	Top-10 Acc.	Mean Positive Rank↓	MRR	Pairwise Acc. (%)
Qwen3-Embedding-8B	–	No	38.38	60.19	74.58	83.40	90.55	98.60	2.71	0.59	87.75
Qwen3-Embedding-8B	–	Default	43.26	66.41	80.30	88.03	93.20	98.56	2.43	0.63	89.73
Qwen3-Embedding-8B	–	MIR-specific	42.59	64.75	79.31	88.06	93.01	99.20	2.44	0.63	89.67
Qwen3-Embedding-8B-FT	No	No	82.50	93.71	97.35	98.69	99.36	100.00	1.29	0.90	97.90
Qwen3-Embedding-8B-FT	No	Default	82.44	93.58	97.25	98.69	99.43	100.00	1.29	0.90	97.90
Qwen3-Embedding-8B-FT	No	MIR-specific	82.47	93.58	97.19	98.72	99.39	99.97	1.30	0.90	97.88
Qwen3-Embedding-8B-FT	Default	No	81.42	93.81	97.32	98.56	99.43	100.00	1.30	0.89	97.84
Qwen3-Embedding-8B-FT	Default	Default	82.02	93.84	97.32	98.63	99.36	99.97	1.30	0.90	97.87
Qwen3-Embedding-8B-FT	Default	MIR-specific	82.09	93.84	97.29	98.75	99.39	99.97	1.30	0.90	97.88
Qwen3-Embedding-8B-FT	MIR-specific	No	81.77	93.65	97.38	98.91	99.39	99.97	1.30	0.90	97.89
Qwen3-Embedding-8B-FT	MIR-specific	Default	82.47	93.90	97.32	98.88	99.52	99.97	1.28	0.90	97.96
Qwen3-Embedding-8B-FT	MIR-specific	MIR-specific	82.34	93.84	97.41	98.75	99.52	99.97	1.29	0.90	97.94
gte-Qwen2-7B-instruct	–	No	38.28	60.09	74.36	83.84	89.78	98.44	2.73	0.59	87.61
gte-Qwen2-7B-instruct	–	Default	44.13	65.55	79.09	86.88	91.35	98.53	2.51	0.63	89.21
gte-Qwen2-7B-instruct	–	MIR-specific	40.90	62.74	76.92	86.62	92.72	99.30	2.51	0.61	89.16
gte-Qwen2-7B-instruct-FT	No	No	74.43	89.02	93.84	95.95	97.64	99.23	1.56	0.84	95.95
gte-Qwen2-7B-instruct-FT	No	Default	74.43	88.98	93.87	96.10	97.83	99.30	1.56	0.84	96.00
gte-Qwen2-7B-instruct-FT	No	MIR-specific	73.15	88.38	93.55	96.30	97.92	99.30	1.57	0.84	95.89
gte-Qwen2-7B-instruct-FT	Default	No	72.83	88.22	93.77	95.59	97.32	99.14	1.61	0.84	95.65
gte-Qwen2-7B-instruct-FT	Default	Default	74.94	89.14	93.93	95.88	97.06	99.01	1.59	0.85	95.78
gte-Qwen2-7B-instruct-FT	Default	MIR-specific	73.31	88.76	93.84	95.63	97.13	99.20	1.60	0.84	95.69
gte-Qwen2-7B-instruct-FT	MIR-specific	No	70.69	86.81	92.56	95.08	96.87	99.30	1.65	0.82	95.33
gte-Qwen2-7B-instruct-FT	MIR-specific	Default	71.90	87.45	93.07	95.69	97.16	99.30	1.61	0.83	95.60
gte-Qwen2-7B-instruct-FT	MIR-specific	MIR-specific	72.38	87.80	93.30	95.82	97.41	99.14	1.61	0.83	95.60
Qwen3-Reranker-8B	–	Default	52.97	74.94	87.13	92.46	95.66	99.30	2.05	0.71	92.50
Qwen3-Reranker-8B	–	MIR-specific	55.81	78.19	88.19	93.42	96.49	99.62	1.94	0.73	93.30
Qwen3-Reranker-8B-FT	Default	Default	86.33	95.82	98.02	99.07	99.55	100.00	1.22	0.92	98.49
Qwen3-Reranker-8B-FT	Default	MIR-specific	86.33	96.01	98.12	99.11	99.52	100.00	1.21	0.92	98.47
Qwen3-Reranker-8B-FT	MIR-specific	Default	86.88	95.98	98.08	99.04	99.39	99.97	1.21	0.93	98.49
Qwen3-Reranker-8B-FT	MIR-specific	MIR-specific	86.91	96.07	97.99	99.14	99.49	100.00	1.21	0.93	98.51
BGE-Reranker-v2-Gemma	–	Default	42.43	64.72	78.45	86.75	91.19	98.63	2.55	0.62	89.01
BGE-Reranker-v2-Gemma	–	MIR-specific	40.23	62.80	76.09	84.39	89.24	97.77	2.72	0.60	87.67
BGE-Reranker-v2-Gemma-FT	Default	Default	82.38	94.13	97.13	98.34	99.17	99.90	1.30	0.90	97.85
BGE-Reranker-v2-Gemma-FT	Default	MIR-specific	80.27	93.23	96.90	98.44	99.11	99.87	1.33	0.89	97.60
BGE-Reranker-v2-Gemma-FT	MIR-specific	Default	81.23	93.36	96.78	98.05	98.98	99.90	1.33	0.89	97.63
BGE-Reranker-v2-Gemma-FT	MIR-specific	MIR-specific	80.24	92.88	96.55	97.80	98.85	99.90	1.35	0.88	97.47

Table 17: Results with different training objectives. Top-1 accuracy (%) is tested across candidate distributions including no background during training and inference, and Proposed Rank refers to mean rank over a pool of 3,035 candidates. FT: Fine-Tuned.

Model	Objective	Original	All Random	All Hard	Decoy Cluster	Proposed Rank↓	No Background
Qwen3-Reranker-0.6B-FT	Pointwise	79.73	85.60	82.69	88.22	56.70	74.01
Qwen3-Reranker-0.6B-FT	Listwise	81.13	87.20	83.62	89.97	50.12	78.93
Qwen3-Reranker-4B-FT	Pointwise	84.29	89.50	86.91	91.63	38.94	80.24
Qwen3-Reranker-4B-FT	Listwise	85.63	91.16	87.80	93.01	34.59	84.55
Qwen3-Reranker-8B-FT	Pointwise	85.57	90.55	87.80	91.89	35.22	81.45
Qwen3-Reranker-8B-FT	Listwise	86.33	91.41	88.31	93.52	31.85	85.25
BGE-Reranker-v2-Gemma-FT	Pointwise	81.80	88.15	83.81	90.45	48.08	78.38
BGE-Reranker-v2-Gemma-FT	Listwise	82.38	88.41	84.74	90.71	45.75	80.08

D INDEXING, CASCADE CONFIGURATION, AND COMPUTATIONAL COST

D.1 INDEPENDENT VERSUS QUERY-CONDITIONED CANDIDATE ENCODING

Table 18 provides detailed results of query-conditioned candidate encoding on the Original candidate set for four embedding backbones, before and after listwise full fine-tuning. Reported metrics include Top-k accuracy, mean positive rank, MRR, and pairwise accuracy.

Table 18: Independent (w/o) and query-conditioned (w/) candidate encoding on the Original test set. Context indicates whether the query is concatenated with the candidate during candidate encoding. -FT denotes listwise full fine-tuning.

Models	Context	Top-1 Acc.	Top-2 Acc.	Top-3 Acc.	Top-4 Acc.	Top-5 Acc.	Top-10 Acc.	Mean Positive Rank↓	MRR	Pairwise Acc. (%)
Qwen3-Embedding-0.6B	w/o	35.09	55.84	70.11	79.82	86.14	97.83	3.01	0.56	85.59
Qwen3-Embedding-0.6B	w/	29.47	49.58	62.64	71.81	78.61	95.40	3.61	0.50	81.27
Qwen3-Embedding-0.6B-FT	w/o	77.39	91.38	95.50	97.89	98.85	99.97	1.40	0.87	97.10
Qwen3-Embedding-0.6B-FT	w/	81.80	93.49	96.87	98.31	98.98	100.00	1.31	0.89	97.75
Qwen3-Embedding-4B	w/o	37.61	58.65	73.08	82.95	90.07	98.79	2.76	0.58	87.42
Qwen3-Embedding-4B	w/	36.56	56.80	69.25	77.78	82.92	96.42	3.20	0.56	84.25
Qwen3-Embedding-4B-FT	w/o	82.12	93.90	97.57	98.56	99.52	100.00	1.29	0.90	97.94
Qwen3-Embedding-4B-FT	w/	86.02	95.88	97.99	99.07	99.49	100.00	1.22	0.92	98.43
Qwen3-Embedding-8B	w/o	38.38	60.19	74.58	83.40	90.55	98.60	2.71	0.59	87.75
Qwen3-Embedding-8B	w/	43.49	64.46	78.10	86.08	91.09	98.66	2.55	0.63	88.93
Qwen3-Embedding-8B-FT	w/o	82.50	93.71	97.35	98.69	99.36	100.00	1.29	0.90	97.90
Qwen3-Embedding-8B-FT	w/	86.85	96.10	98.12	99.04	99.58	100.00	1.21	0.93	98.51
gte-Qwen2-7B-instruct	w/o	38.28	60.09	74.36	83.84	89.78	98.44	2.73	0.59	87.61
gte-Qwen2-7B-instruct	w/	32.12	48.66	60.44	69.19	74.74	91.92	3.98	0.51	78.69
gte-Qwen2-7B-instruct-FT	w/o	74.43	89.02	93.84	95.95	97.64	99.23	1.56	0.84	95.95
gte-Qwen2-7B-instruct-FT	w/	80.72	92.94	96.52	98.08	98.98	100.00	1.34	0.89	97.59

D.2 SHORTLIST SIZE AND CASCADE PERFORMANCE

Table 19 provides results for four retrieve–then–rerank pipelines with shortlist sizes $K \in \{2, 3, 4, 5, 10\}$. It reports Top-1 accuracy on Original, All Random, All Hard, and Decoy Cluster, together with mean positive rank on the All-Positive pool.

Table 19: Cascade performance across shortlist sizes and candidate distributions. Original, All Random, All Hard, and Decoy Cluster report Top-1 accuracy (%); Proposed Rank reports mean positive rank in the 3,035-paper candidate pool (lower is better).

Models	Shortlist K	Original	All Random	All Hard	Decoy Cluster	Proposed Rank↓
Qwen3-Embedding-0.6B-FT → Qwen3-Reranker-0.6B-FT	2	80.62	88.25	82.57	90.20	49.46
Qwen3-Embedding-0.6B-FT → Qwen3-Reranker-0.6B-FT	3	81.29	87.84	83.30	90.17	49.48
Qwen3-Embedding-0.6B-FT → Qwen3-Reranker-0.6B-FT	4	81.26	87.45	83.37	90.13	49.51
Qwen3-Embedding-0.6B-FT → Qwen3-Reranker-0.6B-FT	5	81.16	87.29	83.40	90.07	49.55
Qwen3-Embedding-0.6B-FT → Qwen3-Reranker-0.6B-FT	10	81.10	87.16	83.59	89.97	49.64
Qwen3-Embedding-4B-FT → Qwen3-Reranker-4B-FT	2	85.82	91.03	87.68	93.26	41.19
Qwen3-Embedding-4B-FT → Qwen3-Reranker-4B-FT	3	86.27	91.22	88.09	93.01	41.19
Qwen3-Embedding-4B-FT → Qwen3-Reranker-4B-FT	4	86.08	91.12	88.03	93.07	41.19
Qwen3-Embedding-4B-FT → Qwen3-Reranker-4B-FT	5	85.79	91.19	87.99	93.07	41.20
Qwen3-Embedding-4B-FT → Qwen3-Reranker-4B-FT	10	85.60	91.12	87.87	93.04	41.17
Qwen3-Embedding-8B-FT → Qwen3-Reranker-8B-FT	2	85.89	91.73	87.74	93.49	40.13
Qwen3-Embedding-8B-FT → Qwen3-Reranker-8B-FT	3	86.49	91.70	88.22	93.55	40.14
Qwen3-Embedding-8B-FT → Qwen3-Reranker-8B-FT	4	86.46	91.63	88.41	93.58	40.14
Qwen3-Embedding-8B-FT → Qwen3-Reranker-8B-FT	5	86.43	91.54	88.54	93.58	40.15
Qwen3-Embedding-8B-FT → Qwen3-Reranker-8B-FT	10	86.33	91.51	88.35	93.52	40.14
gte-Qwen2-7B-instruct-FT → BGE-Reranker-v2-Gemma-FT	2	80.49	87.36	82.85	89.37	92.15
gte-Qwen2-7B-instruct-FT → BGE-Reranker-v2-Gemma-FT	3	81.80	87.52	84.36	89.91	92.14
gte-Qwen2-7B-instruct-FT → BGE-Reranker-v2-Gemma-FT	4	82.12	87.90	84.64	90.07	92.12
gte-Qwen2-7B-instruct-FT → BGE-Reranker-v2-Gemma-FT	5	82.28	87.87	84.83	90.13	92.11
gte-Qwen2-7B-instruct-FT → BGE-Reranker-v2-Gemma-FT	10	82.38	88.03	84.67	90.26	91.97

D.3 EMBEDDING COMPRESSION AND DIMENSION SENSITIVITY

Figure 3 evaluates different embedding dimensions. Substantial accuracy can be retained with lower-dimensional representations in the tested setting. This does not prove constant sign-rank for unseen queries or arbitrarily large corpora.

Table 20: Inference efficiency on 3,132 queries using a single H200 GPU. Latency and throughput are reported as mean ± sample standard deviation over three runs; FLOPs/query is reported as a mean. Model loading and warm-up are excluded. Cascade latency is obtained by summing separately measured embedding and reranking times.

Method	Setting	Mean Latency BS=1 (ms) ↓	Throughput BS=1 (q/s) ↑	FLOPs/query (10^{12}) ↓
MS-IR-7B	N/A	$24{,}700.00 \pm 70.00$	0.04049 ± 0.00011	349.62
DeepSeek-R1-Distill-Qwen-7B-Embedding-FT	Pre-indexed	18.44 ± 0.08	54.24250 ± 0.22820	6.30
DeepSeek-R1-Distill-Qwen-7B-Embedding-FT	Strict cold	244.81 ± 1.00	4.08480 ± 0.01660	76.13
DeepSeek-R1-Distill-Qwen-7B-Reranker-FT	15 candidates	414.16 ± 0.50	2.41450 ± 0.00290	179.92
DeepSeek-R1-Distill-Qwen-7B-Embedding-FT (Top-3) → DeepSeek-R1-Distill-Qwen-7B-Reranker-FT	Pre-indexed	102.82 ± 0.90	9.72580 ± 0.08440	42.32

Table 21: Inference efficiency on 3,132 queries using a single H200 GPU. Latency and throughput are reported as mean ± sample standard deviation over three runs; FLOPs/query is reported as a mean. Model loading and warm-up are excluded. Cascade latency is obtained by summing separately measured embedding and reranking times.

Method	Setting	Mean Latency BS=1 (ms) ↓	Throughput BS=1 (q/s) ↑	FLOPs/query (10^{12}) ↓
MS-IR-7B	N/A	$24{,}700.00 \pm 70.00$	0.04049 ± 0.00011	349.62
DeepSeek-R1-Distill-Qwen-7B-Embedding-FT	Pre-indexed	18.44 ± 0.08	54.24250 ± 0.22820	6.30
DeepSeek-R1-Distill-Qwen-7B-Embedding-FT	Strict cold	244.81 ± 1.00	4.08480 ± 0.01660	76.13
DeepSeek-R1-Distill-Qwen-7B-Reranker-FT	15 candidates	414.16 ± 0.50	2.41450 ± 0.00290	179.92
DeepSeek-R1-Distill-Qwen-7B-Embedding-FT (Top-3) → DeepSeek-R1-Distill-Qwen-7B-Reranker-FT	Pre-indexed	102.82 ± 0.90	9.72580 ± 0.08440	42.32

D.4 Efficiency Measurement Protocol

We evaluate 3,132 queries on a single H200 GPU over three runs, excluding model loading and warm-up. For each run, mean time per query is accounted evaluation time divided by query count; throughput is query count divided by that time. We report the mean and sample standard deviation of these run-level quantities. Table 20 uses milliseconds throughout. The MS-IR row uses the post-hoc accounting described below rather than unadjusted wall-clock time.

D.5 Detailed Runtime and Compute Accounting

Pre-indexed results exclude offline candidate-embedding construction from online query time. Strict cold denotes online candidate re-encoding, not model loading. The reranker scores 15 candidates per query. Cascade costs sum separately measured query-embedding and Top-3-reranking times within each repeat; they are not a single end-to-end pipeline measurement. FLOPs/query is a theoretical operation-count estimate, reported as a mean in units of 10^{12}, rather than measured hardware utilization.

E Hypothesis Evaluation Protocols and Qualitative Examples

E.1 Controlled Hypothesis Generation

The downstream comparison varies the inspiration source among random selection, MS-IR, the learned GTE–BGE Top-3 cascade, and the annotated inspiration. The hypothesis generator and generation settings are held fixed across sources as described in the main experiment. The comparison concerns inspiration selection within this pipeline, not changes to the generator.

E.2 LLM Judging Rubric and Evaluation Procedure

The LLM evaluation ranks the four hypotheses for each problem, with lower rank indicating a better assessment. Table 3 reports mean ranks over the test set. These are relative judgments among compared outputs, not calibrated probabilities of scientific correctness or experimental success.

E.3 Expert Assessment and Annotation Procedure

The expert evaluation uses 50 randomly sampled test instances and three evaluators. Table 3 reports each evaluator's mean rank and their average. This sample provides a complementary human assessment, not expert validation of every generated hypothesis.

Table 22: Alternative explanations

Hypothesis	Test	Result
Key prior (any good key helps)	Shuffle neighbour keys in kNN transfer	$16.1\% \rightarrow 0.01\%$
Format only	Train on deranged positives	Top-1 12.2% (base 35.1%), two protocols
Topic prior	All-Hard candidates (same-topic negatives)	Base $34.4 \rightarrow$ 2k random $69.3 \rightarrow$ FT 79.9
Memorized pairs	Recovery at coverage below the lowest full-data decile	54.9% $[52.3,\ 57.4]$ with 2k random rows
Text matching of h_{j-1}	Multi-step papers, All-Positive top-1	Re-selects used inspiration $44.9\% \rightarrow 23.4\%$; step-1 current-vs-future $45.5\% \rightarrow 58.8\%$
Lexical leakage	BM25 over the same candidates	$30.9\% <$ base $35.1\% \ll$ FT; 4-gram overlap 0.15%

E.4 Detailed Evaluation Results and Agreement Analysis

The expert-average ordering in Table 3 places the annotated-inspiration and learned-retrieval conditions ahead of MS-IR and random selection. Individual evaluators do not give identical orderings. Average ranks alone do not quantify inter-rater agreement or establish statistical significance; neither is inferred from this table.

F Limitations

F.1 Incomplete Annotations and Alternative Inspirations

The annotated inspiration need not be the only useful paper for a context. Accuracy measures recovery of this annotation and can penalize valid alternatives. The annotated-inspiration condition is consequently a reference, not a guarantee of the best possible inspiration.

F.2 Limits of the Pattern-Matching Interpretation

The results are consistent with reusable problem–inspiration matching patterns but do not uniquely identify the model's internal mechanism. Finite-set representability and empirical dimension saturation are distinct from generalization to unseen scientific problems. Neither rules out dataset-specific regularities.

F.3 Generalization beyond Evaluated Settings

The experiments cover the reported temporal split, backbones, and candidate distributions. Controlled pools and the 3,035-paper pool do not reproduce all challenges of a changing, large-scale scientific corpus. A temporal split alone does not establish the absence of pretraining contamination or all forms of information leakage.

F.4 Need for Prospective Scientific Validation

The study evaluates retrospective retrieval and generated hypotheses. LLM and expert rankings assess perceived quality, not experimental validity. Establishing new discoveries requires prospective testing, domain-specific feasibility checks, and empirical validation beyond the reported evaluations.