أكاديمية كاوست
KAUST ACADEMY

جامعة الملك عبدالله
للعلوم والتقنية
King Abdullah University of
Science and Technology

Can We Factorize Text?
LLMs as Training-Free Subquadratic Retrievers

Anas Obayd, Raed Rashid, Abdulrahman Alghamdi,
Abdulrahman Mousa, Abdullah Mahmoud, Mohamed Eltahir,
Tanveer Hussain

anas.obayd@kaust.edu.sa, raed.rashid@kaust.edu.sa, abdulrahman.alghamdi@kaust.edu.sa,
abdulrahman.mousa@kaust.edu.sa, abdallah.ahmed@kaust.edu.sa, mohamed.hamid@kaust.edu.sa,
htanveer3797@gmail.com

Abstract

Large Language Models (LLMs) are good at judging relevance but have a quadratic attention cost, and a limited context window. **CorpusProbe** factorizes the document to reduce the context window while also producing a subquadratic computational cost. We factorize a document into K chunks and pass through 2K yes/no single-pass probes, with the first K covering local context and the second K covering global context. Tests on the full corpora of eight BRIGHT long-documents, CorpusProbe achieved the highest NDCG@10 (0.526) and best AUC (0.976) in comparison to six other methods, including naively passing the full document to an LLM (0.967 vs 0.753). The main reason for this disparity was attributed to the LLM being unable to fit the whole document in its context window without crashing. Finally, we tested whether CorpusProbe's cell scores can accurately localize relevant information within documents via manually annotating 100 tip-of-the-tongue documents from OBLIQ-Bench and found that it outperforms the dedicated ColBERT and frontier-embedding models (50.5 vs 40.0 vs 40.0).

Date: August 24, 2026
Code: https://github.com/Pineyapple/LLM-as-a-retriever
Project Page: https://github.com/Pineyapple/LLM-as-a-retriever

Contents

1 Introduction 2

2 Related Work 2

3 Method: Factorized Text Scoring 3

3.1 Notations . . . 3

3.2 Reading strategies . . . 3

3.3 Evidence localization at zero additional cost . . . 4

3.4 Cost model . . . 4

4 Experiments 4

4.1 Setup . . . 5

4.2 Main results . . . 5

4.3 Feasibility . . . 5

4.4 Evidence localization . . . 6

5 Analysis 7

5.1 Similarity and verdicts fail differently . . . 7

5.2 Whole-document reading is high-variance . . . 7

5.3 The wall is not a uniform tax . . . 7

5.4 Where each reading wins . . . 8

5.5 Why strided probes matter . . . 8

5.6 What the probe is, and is not . . . 8

5.7 Cost, stated plainly . . . 9

6 Limitations and Future Work 9

7 Conclusion 10

References 10

A Implementation Details 11

B Cost Measurements 12

C Full Per-Domain Results 12

D Localization Protocol 12

1 Introduction

A frozen LLM can accurately judge relevance with a given query. The issue is that it has a quadratic computational cost and a monolithic memory footprint. On BRIGHT long-document benchmark, an LLM failed to read 0.7-26% of the corpus at a 16,384 token budget, while our method read all of them on the same hardware. This paper aims to answer: *can the text be factorized?* What is lost by doing so? We propose adapting GridProbe (cite) to text. **CorpusProbe** covers the document's chunks with 2K yes/no probes (where $K = sqrt(N/c)$ token/chunk_size): K *local* probes, each over K consecutive chunks, and K *strided* probes, each spanning the whole document by sampling every K-th chunk. The LLMs scores are read from the answer space to ensure a single forward pass. The attention cost falls from N^2 to $N^1.5$, but the linear term's coefficient is doubled due to the increase in calls, so the method only wins over a standard LLM past a certain document size. Ultimately, the method is cheaper asymptotically. Yet little is lost, CorpusProbe outperforms naively reading over the whole document on **eight BRIGHT domains** (0.967 vs 0.753 AUC). Though, on the corpus of short documents, whole-document reading wins. The result of local and global context's outer-product also finds the relevant text within the document at no added cost, performing better than the dedicated ColBERT.

Our contributions:

- **Factorized text scoring**, a training-free reading strategy for frozen LLM relevance judgments that is bounded per call, subquadratic in document length, and runs unchanged past the context wall where whole-document reading fails.
- **The best NDCG@10 and AUC of six methods** on full-corpus retrieval across eight BRIGHT domains, with three controlled pairwise comparisons that isolate the mechanism: versus whole-document reading (AUC 8/8), versus a parameter-matched embedder (NDCG@10 7/8), and versus local-only probing at identical granularity (AUC 7/8).
- **A characterization of whole-document reading** as high-variance and, on one domain, anti-correlated with relevance — together with the regime boundary showing where it remains the better choice.
- **Evidence localization for free**: the ranking score decomposes into passage-level evidence that outperforms dedicated localization baselines.
- **An honest cost account**: closed-form $s^{1.5}$ scaling with measured crossovers, and the explicit concession that below the wall factorized scoring costs a median 5× more FLOPs. We claim quality, feasibility, and interpretability; we do not claim cheap.

2 Related Work

Similarity Retrieval. Lexical matching, such as BM25 provided way to dense dual encoders which led to multi-vector late interaction like ColBERT. Each of these depended on similarity scores between independently evaluated queries and documents.

LLM rerankers. These lie at the end of a RAG system. Methods such as pointwise scoring, listwise generation, and setwise comparison. All of them excel at ranking relevant documents, but they are limited by the first-stage retrievers' results and scale quadratically with document length.

Long documents and truncated reads.. Classical IR bounds the read by decomposition: FirstP scores the lead passage, MaxP the best passage [1]. Both presume relevance is locally concentrated and both discard cross-passage structure. Long-context architectures push the wall outward while keeping quadratic economics and monolithic memory, so the wall moves but the scaling does

Table 1 Comparison of retrieval approaches.

Method	Train-free	Beyond window	Subquad.	Global view	Self-loc.	Full corpus
BM25 / dense retrievers	×	✓	✓	×	×	✓
ColBERT late interaction	×	✓	✓	×	~	✓
LLM rerankers (point/list/set)	~	×	×	✓	×	×
FirstP / MaxP aggregation	×	✓	✓	×	~	×
Long-context whole-document read	✓	×	×	✓	×	✓
CorpusProbe (ours)	✓	✓	✓	✓	✓	✓

not — and extending the window does not by itself improve judgment, since quality degrades as relevant content sits deeper inside a long context [2]. Systems work on reusable document caches lowers the constant of reading without changing its shape [3–5]. *What is required is a way to read all of a document through pieces whose individual cost is bounded, while preserving a document-wide view.*

Verification in the loop.. A related line uses model verification as feedback, iterating a retriever until the model judges its output sufficient [6]. There the verifier steers an index that still makes the relevance decision. Our use of verification is different in kind: the verdict itself is the score. *What is required is a change to the deciding mechanism, not a wrapper around it.*

Explanation and attribution.. Passage-level explanation is normally a second system: highlighting, rationale extraction, or a separate chunk-level encoder run over a document that something else retrieved. This costs extra computation and carries no guarantee that the highlighted passage is what drove the ranking score. *What is required is a score that is its own explanation.*

Where CorpusProbe stands.. CorpusProbe meets these five requirements at once. It decides relevance after the query is known, by verdict rather than distance. Its factorized probes cost $s^{1.5}$ compute with probe-bounded memory, so document length never forces a truncation. It reads every chunk of arbitrarily long documents through short bounded probes — local and strided — composed into one global judgment. It changes the deciding mechanism rather than wrapping an index. And its score decomposes, by construction, into the passage-level evidence that produced it. Table 1 states the resulting position; each column is a requirement one of the blocks above leaves unmet.

3 Method: Factorized Text Scoring

3.1 Notations

A corpus C contains D documents. Given a query q, the system returns a ranking over documents and, on request, the passage inside a document that justifies its rank. We write s for the token length of a document, p for the length of the query prompt, and c for the chunk size (100 tokens throughout).

Every score in this paper is produced identically. A frozen instruction-tuned model receives a query and a piece of text inside a fixed template ending in `"Is this document relevant to the query? Answer Yes or No:"`, and we read the logits

3.2 Reading strategies

Split a document into $N = s/c$ chunks $c_1, \ldots, c_N$ in reading order. The strategies we compare are points on a single axis — how much of the document, and in what arrangement, each call sees:

- **FullRead**: one call over the whole document. Undefined when $s + p$ exceeds the context budget; such documents cannot be scored at all.
- **FirstP**: one call over c_1 alone [1]. The cheapest possible read and the degenerate endpoint of the family.
- **LocalOnly**: $K = \lceil \sqrt{N} \rceil$ calls over all the local chunk rows.
- **CorpusProbe**: LocalOnly's K *local probes* plus K *strided probes*, the j-th joining every K-th chunk starting at c_j. Every chunk is read exactly twice — once with its neighborhood, once inside a document-wide sample — by $2\sqrt{N}$ calls of at most $\sqrt{sc}$ tokens each.

Each cell's score is computed using

$$M[i, j] = i_{\text{score}} \times j_{\text{score}},$$
$$\text{Score}(D) = \max(M).$$

All documents are ranked based on this score, and the top K documents are retrieved.

3.3 Evidence localization at zero additional cost

For a returned document, $M[i, j]$ assigns a score to every chunk: chunk c_k lies at the intersection of exactly one local probe and one strided probe, and M peaks where both agree. At that cell, the most relevant information in the document is located, which can be read to find the relevant text.

3.4 Cost model

For a document with s/c tokens, CorpusProbe makes $2K$ forward calls in total (K row probes and K column probes), giving it a call complexity of $O(\sqrt{s/c})$.

The equation

$$\text{Cost}_{\text{Naive}} = 2Ps + 4Lds^2,$$

represents the approximate TFLOP cost of a single forward pass over the full s-token document.

By chunking, each individual call now receives $n = \sqrt{s/c}$ tokens rather than the full s, across $2K$ such calls, which simplifies to:

$$\text{Cost}_{\text{CorpusProbe}} = 4PN + 8Ld\,K^{1/2}N^{3/2}.$$

This shows CorpusProbe's overall complexity is $O(s^{1.5})$.

1. **Bounded memory.** Per-call memory is $\sqrt{sc}$ rather than s, so no call approaches the wall at *any* document length. This is a change in kind, not degree: the method's feasibility does not depend on document length at all.
2. **Parallelism.** The $2\sqrt{N}$ probes are independent short sequences, so up to $\sqrt{s/c}$ of them run in flight where a monolithic read admits none. This is why the measured latency crossover (~6–14k tokens) arrives an order of magnitude before the FLOPs crossover (~16–66k).

We state the constant plainly. Every chunk is read twice, so below the crossover the factorized read costs *more* FLOPs than the whole-document read — a median 5× across our benchmark and up to 27× on its largest corpus. The claims of this paper are quality, feasibility, and interpretability; the compute claim is asymptotic, not per-token.

4 Experiments

Table 2 BRIGHT `long_documents`, 8 domains, **full corpus, no pooling, no gold injection**, 10 queries per domain, judge Qwen3-4B-Instruct-2507. BM25 and Qwen3-Embedding-4B are rerun under this identical protocol. *Read%* is the fraction of documents the method can process at a 16,384-token budget. FullRead's compute is the price of scoring only the documents it can read.

Method	NDCG@10	AUC	Read%	TFLOPs/query
similarity				
BM25	.203	.831	100	≈0
Qwen3-Embed-4B (param-matched)	.390	.953	100	index lookup
LLM reading, one frozen 4B judge				
FirstP (first chunk only)	.246	.791	100	**898–5,536**
FullRead (whole document)	.485	.753	**73.9–99.3**	5,324–35,702
LocalOnly ($\sqrt{N}$ local probes)	.475	.953	100	6,720–484,074
CorpusProbe (local + strided)	**.526**	**.967**	100	13,706–969,097

4.1 Setup

Benchmark and protocol.. We evaluate on BRIGHT `long_documents` [7], eight domains of reasoning-intensive queries. Document lengths span a median of 334–6,702 tokens by domain with maxima above two million. We report NDCG@10, Recall@10, and AUC (the probability a relevant document outranks an irrelevant one, computed over all query–document pairs), and the measured per-query compute.

Systems.. All LLM rows use one frozen judge, Qwen3-4B-Instruct-2507, with the scoring primitive of §3.2 held fixed; the four readings differ only in which text reaches it. **BM25** [8] is the lexical floor. **Qwen3-Embedding-4B** is the parameter-matched similarity control. Documents that FullRead cannot fit are scored as misses.

4.2 Main results

CorpusProbe attains the best NDCG@10 and the best AUC of all six methods.

Versus whole-document reading: AUC in all eight domains.. 0.967 against 0.753, a margin of +0.214, with CorpusProbe ahead in every domain of Table 3. Reading a document in bounded pieces ranks a corpus better than reading it whole.

Versus the parameter-matched embedder: NDCG@10 in seven of eight.. 0.526 against 0.390, +35% relative. At equal parameter count the only difference is the deciding mechanism, so this is the cleanest evidence that the verdict, not the scale, is what produces the gain.

Versus local-only probing: AUC in seven of eight.. LocalOnly and CorpusProbe share chunk size, judge, prompt, and aggregation, and differ *only* by the $\sqrt{N}$ strided probes. The strided probes are therefore isolated as the carrier of document-wide structure — local context around a chunk is not sufficient, and a read that spans the document is what closes the gap.

4.3 Feasibility

FullRead can process only 73.9–99.3% of documents at a 16,384-token budget; every factorized reading processes 100% by construction, because no single call exceeds $\sqrt{sc}$ tokens regardless of document length. The wall is a property of hardware and budget rather than of the benchmark — a companion run on 16GB-class hardware at a smaller budget failed on 29% of documents overall — so we report the budget and hardware alongside the rate rather than a single universal number.

Table 3 Per-domain results, full corpus. *Read%* refers to FullRead. CorpusProbe leads AUC against FullRead in every domain. FullRead's three NDCG@10 wins fall on domains it can most fully read; StackOverflow is the exception analyzed in §5.3.

Domain	Read%	BM25	Embed-4B	FirstP	FullRead	LocalOnly	CorpusProbe
NDCG@10							
biology	86.9	.306	.724	.646	.747	.751	**.872**
earth_science	73.9	.249	.556	.437	.678	.726	**.759**
economics	79.8	.100	.365	.100	.300	.452	**.506**
pony	99.3	.388	.152	.140	**.600**	.327	.301
psychology	79.3	.186	.437	.100	.376	.449	**.532**
robotics	87.2	.088	.259	.200	**.696**	.506	.564
stackoverflow	91.4	.143	**.261**	.063	.000	.185	.216
sustainable_living	89.1	.163	.368	.284	**.482**	.405	.457
mean		.203	.390	.246	.485	.475	**.526**
AUC							
biology		.845	.983	.932	.924	.982	**.991**
earth_science		.910	**.992**	.917	.878	.947	.954
economics		.733	.953	.651	.733	.963	**.969**
pony		**.957**	.909	.868	.941	.933	.951
psychology		.727	.954	.617	.676	.929	**.964**
robotics		.709	.886	.738	.979	**.986**	.983
stackoverflow		.962	**.984**	.827	**.108**	.964	.980
sustainable_living		.804	**.967**	.774	.787	.918	.947
mean		.831	.953	.791	**.753**	.953	**.967**

What does not depend on the budget is the shape of the failure: whole-document reading does not degrade as documents lengthen, it stops.

4.4 Evidence localization

The probe scores that rank a document also explain it (§3.3). To measure this we manually annotated 100 documents, marking for each query the chunks containing the evidence that justifies retrieval, and where several were relevant, which was most relevant. We compare against two systems built specifically for passage-level scoring: ColBERT [9] and a frontier embedding model, each run over the same chunking.

CorpusProbe leads on every metric (Table 4). The clearest separation is at top-1: it points at an annotated span for 58% of queries against 46% and 43%, over a 29.6% chance floor. Because every method saw identical queries, the comparison is paired: against the frontier embedder CorpusProbe localizes 31 queries the baseline misses while losing 16 (exact McNemar $p = 0.040$); against ColBERT the split is 26 versus 14 ($p = 0.081$). The architectural difference compounds the accuracy one — the baselines require a separate chunk-level pass over a document that some other system must first retrieve, whereas CorpusProbe's map falls out of the scores that did the retrieving.

Table 4 Evidence localization on 100 manually annotated documents, given the correct document. Grids are small (median 14 chunks, ~4 annotated), so the chance row is essential context: CorpusProbe's top-1 accuracy is 2.0× chance. *Graded* awards partial credit when the most-relevant chunk appears at ranks 2–4. CorpusProbe's map is a by-product of the ranking computation; both baselines require a separate scoring pass.

Method	Top-1	P@3	MRR	Graded
Chance (median 14 chunks)	29.6	29.6	—	—
ColBERT	46.0	42.3	62.2	69.0
Frontier embedding	43.0	40.7	61.8	72.0
CorpusProbe (free)	**58.0**	**48.0**	**71.6**	**74.4**

5 Analysis

5.1 Similarity and verdicts fail differently

The parameter-matched embedder nearly ties CorpusProbe on AUC (0.953 versus 0.967) yet trails it by 35% relative on NDCG@10 (0.390 versus 0.526). The two metrics measure different things: AUC asks whether relevant documents are generally above irrelevant ones, NDCG@10 asks whether they are at the very top. Similarity is good at the former and weak at the latter — it places relevant documents in roughly the right region of a ranking, and the verdict is what resolves the head of the list, which is where retrieval is actually consumed. This dissociation is the central thesis of the paper appearing as a measurement rather than an assertion.

5.2 Whole-document reading is high-variance

The mirror dissociation holds for FullRead, and it runs the other way: second-best NDCG@10 (0.485) but the *worst* AUC of all six methods (0.753, below BM25's 0.831). One long read is precise when it succeeds and catastrophic when it fails. Excluding StackOverflow its AUC is 0.845, so a single domain drives much of the gap — but that domain is not an outlier to be discarded, it is the clearest instance of the failure mode.

5.3 The wall is not a uniform tax

On StackOverflow, FullRead attains AUC 0.108: its ranking is *anti-correlated* with relevance, worse than random. Every factorized reading recovers the domain completely (LocalOnly 0.964, CorpusProbe 0.980), and NDCG@10 moves from exactly zero to 0.216.

Overall readability does not explain this. FullRead reads 91.4% of StackOverflow, the third-highest rate in the benchmark, and reads far less of domains where it performs respectably. What distinguishes StackOverflow is *which* documents it cannot fit. On a programming forum the long, heavily detailed answers are disproportionately the relevant ones, so a reader that skips whatever exceeds its context skips the answers preferentially rather than uniformly. A method whose failures are correlated with relevance does not merely lose accuracy; it inverts. Factorized reading has no wall to select on, which is why the same judge, the same prompt, and the same chunking recover the domain entirely by changing only how the text is delivered. We report the inversion as measured and this account of it as the explanation the evidence supports; isolating the correlation quantitatively across domains is left to future work.

5.4 Where each reading wins

FullRead wins NDCG@10 on pony, robotics, and sustainable_living — three of the four domains with the highest fraction of readable documents (99.3%, 87.2%, 89.1%) — and loses on all four domains where it can read least (86.9% down to 73.9%). Its quality advantage and its feasibility are not independent properties: whole-document reading is competitive precisely while it can still read the corpus. StackOverflow is the informative exception, and it sharpens rather than weakens the rule: with 91.4% readability it should be in FullRead's favourable regime, and it is its worst result, because the governing variable is not how many documents a method fails on but whether those failures land on the relevant ones.

Two further boundaries are worth stating. On pony — the shortest-document domain (median 334 tokens, 99.3% readable) — FullRead beats CorpusProbe outright and BM25 (.388) beats the neural embedder (.152): there is little to factorize and lexical matching works. And FirstP, reading only the first 100 tokens, reaches AUC 0.791 at 10–170× less compute than CorpusProbe, which makes it a strong cheap baseline even though its NDCG@10 of 0.246 shows a fragment cannot resolve the top of a ranking.

5.5 Why strided probes matter

LocalOnly and CorpusProbe share granularity, judge, prompt, and aggregation and differ only by the strided probes; CorpusProbe wins AUC in seven of eight domains. Local context around a chunk is not enough, and some read must span the document. This is factorization's answer to the observation that long-context quality degrades as relevant content sits deeper inside the window [2]: rather than asking one call to hold a long context without losing its middle, we issue many bounded calls, two of which pass through every chunk.

5.6 What the probe is, and is not

Three properties of logit-probing bear on how these results should be read, and we report them because they constrain the method rather than flatter it.

Terse prompts win.. Across five prompt variants evaluated on a held-out development split, the plain control (`"Is this document relevant to the query? Answer Yes or No:"`) outperformed every elaborate alternative, including a task-aware preamble instructing the model about latent relevance and a reframing as a property-checking question. We therefore fixed one prompt before running any headline experiment rather than tuning per benchmark.

The probe is not invariant under negation.. Asking the logically equivalent negated question and reading the opposite token destroys most of the signal, with both answer tokens verified present in the readable distribution throughout, so this is a property of the model's behaviour rather than an artifact of the readout. The mechanism is a loss of decisiveness: the score distribution compresses toward indifference and the spread that carries ranking information disappears. A verdict that does not survive restatement is a genuine limitation of reading relevance off a single logit.

The rejection mass is near-binary.. Scores concentrate on a small number of distinct values among documents the model confidently rejects, so relative positions deep inside the rejected mass carry little information. This is why we report recall at depth rather than median rank for buried documents, and why AUC — computed over all pairs and robust to ties — is our stable metric.

5.7 Cost, stated plainly

Below the context wall CorpusProbe costs 2.6–27× FullRead's FLOPs (median ≈ 5×), because every chunk is read twice. We do not claim a compute advantage in that regime. Three qualifications belong beside that number. FullRead's cost is the price of scoring only 74–99% of a corpus, since it silently skips what it cannot read. The latency picture is more favourable than the FLOPs picture because probes batch where a monolithic read cannot, so the measured latency crossover (~6–14k tokens) arrives an order of magnitude before the FLOPs crossover (~16–66k). And past the wall the comparison ceases to be about cost at all.

6 Limitations and Future Work

We are not cheaper below the wall.. Every chunk is read twice, so below the crossover the factorized read costs more FLOPs than the whole-document read — a median 5× on our benchmark. The claims of this paper are quality, feasibility, and interpretability; a practitioner optimizing FLOPs on short documents should read documents whole, and our own length router says so.

Query samples.. Ten queries per domain (80 in total) keeps full-corpus scoring affordable but leaves per-domain NDCG@10 noisy, which is why our claims rest on AUC — computed over every query–document pair, and stable where a small query count makes top-k metrics swing. Scaling the query count on the cheaper domains is straightforward and is in progress.

The short-document regime.. On the corpus of shortest documents (median 334 tokens, 99.3% readable), whole-document reading wins NDCG@10 outright and lexical matching beats the neural embedder. There is little to factorize when a document fits comfortably in one call, and the method's habitat is long documents.

Corpus scale.. Scoring cost is linear in the number of documents scored. Full-corpus scoring is practical at the scale evaluated here (500–1,900 documents per domain); at substantially larger D a first stage must prune, and the resulting recall ceiling becomes the binding constraint rather than the reading strategy. Amortizing probe prefills into a reusable key-value index [3–5] would collapse per-query cost toward the query length, since every probe's text is query-independent; this is designed but not yet measured end to end.

Judge dependence.. Scoring quality rides the underlying model, and preliminary experiments at larger scale improve it consistently, so our numbers should be read as a floor rather than a ceiling for the method.

Probe reliability.. As §5.6 reports, probe scores move under paraphrase and are not invariant under logical negation. We fixed a single prompt before running any headline experiment and report sensitivity rather than tuning around it, but prompt-robust probe design remains open.

Future work

Four directions follow directly from what is measured here.

Retrieval where relevance is latent.. The strongest motivation for factorized scoring is not long documents alone but queries whose relevance has no surface expression — where similarity retrieval is structurally blind and a verdict is the only mechanism that can decide. We are extending this evaluation to oblique retrieval [10], where the natural question is how much of a corpus should receive a verdict: because a factorized verdict is a batch of short bounded

calls, the retriever-versus-reranker distinction becomes a tunable coverage dial rather than an architectural commitment, and its optimum is an empirical property of the corpus and the judge. Preliminary measurements on that benchmark indicate that verdicts recover substantial recall that an embedding first stage discards, and that the binding constraint on top-k quality is the judge's precision rather than how much of the corpus is covered — a full study is in progress.

The stride knob.. Sampling every t-th probe interpolates continuously between FirstP and the full complement, making the fraction of a document that reaches the judge a tunable budget. Preliminary cross-model measurements suggest a substantial cost reduction at little quality loss; charting that frontier on a single judge would turn factorized scoring from one method into a family with an explicit quality–budget dial.

Amortized serving.. A key-value index over probe prefills, combined with the natural batchability of short probes, would measure the end-to-end throughput story that the cost model predicts but that our per-document accounting cannot capture.

Attribution-grade retrieval.. Because the ranking score decomposes into the passage that produced it, every result can ship with its evidence at no additional cost. For corpora where the passage *is* the product — contracts, transcripts, regulatory filings — this turns retrieval output into an auditable artifact, and distilling those maps into training signal for cheaper retrievers is a natural next step.

7 Conclusion

Can we factorize text? The judgment of a frozen language model survives factorized reading and improves under it. Covering a document with bounded local and strided probes ranks a corpus better than reading documents whole in every domain we measured (AUC 0.967 versus 0.753), beats a parameter-matched embedder where rankings are actually consumed (+35% NDCG@10), and beats local-only probing at identical granularity, isolating document-wide reads as the component that matters. It does all of this while running unchanged at any document length, where whole-document reading does not merely slow down but stops — and on one domain does worse than stopping, ranking anti-correlated with relevance because the documents it cannot fit are the ones the query was looking for. The same probes that produce the ranking also produce its explanation, localizing evidence better than systems built for that purpose, at no additional cost. The price is stated on the label: below the wall, factorized scoring pays a median 5× in FLOPs for its quality, feasibility, and interpretability; past the wall, there is nothing else on the menu. One sentence is worth keeping. A document does not need to be read whole to be judged well — it needs to be asked the right bounded questions, and the answers compose.

References

[1] Zhuyun Dai and Jamie Callan. "Deeper Text Understanding for IR with Contextual Neural Language Modeling". In: *Proceedings of SIGIR*. 2019.

[2] Nelson F. Liu et al. "Lost in the Middle: How Language Models Use Long Contexts". In: *Transactions of the Association for Computational Linguistics* 12 (2024).

[3] Songshuo Lu et al. "TurboRAG: Accelerating Retrieval-Augmented Generation with Precomputed KV Caches for Chunked Text". In: *arXiv preprint arXiv:2410.07590* (2024).

[4] Chao Jin et al. "RAGCache: Efficient Knowledge Caching for Retrieval-Augmented Generation". In: *arXiv preprint arXiv:2404.12457* (2024).

[5] Jiayi Yao et al. “CacheBlend: Fast Large Language Model Serving for RAG with Cached Knowledge Fusion”. In: *arXiv preprint arXiv:2405.16444* (2024).

[6] Xiaonan Li et al. “LLatrieval: LLM-Verified Retrieval for Verifiable Generation”. In: *arXiv preprint arXiv:2311.07838* (2023).

[7] Hongjin Su et al. “BRIGHT: A Realistic and Challenging Benchmark for Reasoning-Intensive Retrieval”. In: *arXiv preprint arXiv:2407.12883* (2024).

[8] Stephen Robertson and Hugo Zaragoza. “The Probabilistic Relevance Framework: BM25 and Beyond”. In: *Foundations and Trends in Information Retrieval* 3.4 (2009), pp. 333–389.

[9] Omar Khattab and Matei Zaharia. “ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT”. In: *arXiv preprint arXiv:2004.12832* (2020).

[10] Diane Tchuindjo, Devavrat Shah, and Omar Khattab. “OBLIQ-Bench: Exposing Overlooked Bottlenecks in Modern Retrievers with Latent and Implicit Queries”. In: *arXiv preprint arXiv:2605.06235* (2026).

A Implementation Details

Scoring.. All probes use a single frozen judge, Qwen3-4B-Instruct-2507, served with vLLM in bfloat16 at a 16,384-token context. Each probe is one forward pass with `max_tokens`= 1; we read the top-20 output log-probabilities at the answer position and compute $\ell = \log P(\texttt{Yes}) - \log P(\texttt{No})$. No text is generated.

Sampling parameters.. We use temperature 1.0 with `top_p`= 1, `top_k`= −1, `min_p`= 0 — deliberately *not* the model card’s generation defaults. Those are tuned for sampling multi-token text, whereas we read one distribution. The distinction matters in one direction only: temperature cannot change a ranking, since dividing every log-odds by the same constant is monotone, but `top_p` and `top_k` are truncation filters, and with the model confidently answering “No” for most documents a nucleus cut can remove `Yes` from the returned distribution and floor the score. We therefore disable truncation and read the raw distribution.

Answer tokens.. The surface form the model emits (`"Yes"` versus `" Yes"`) is model- and template-dependent, and scoring the wrong variant silently introduces a large error while producing no exception. We detect the emitted form empirically at startup from a single dummy probe and log the resulting token id, rather than hardcoding a preference order.

Aggregation.. Maxima are taken with no default floor value. A fixed default (whether 0 or a large negative constant) clamps every score below it onto one identical value, which collapses ordering among rejected documents into a single tie block. We handle the four cases of token presence explicitly so that the per-document floor survives as a tie-breaker.

Chunking.. Documents are tokenized once and split into 100-token chunks; the chunk list is cached per document, since corpora are scored against many queries and re-chunking dominates CPU time otherwise. $K = \lceil \sqrt{N} \rceil$ and the final grid row is padded.

Baselines.. BM25 uses $k_1 = 0.9$, $b = 0.4$. Qwen3-Embedding-4B uses last-token pooling with L_2 normalization and the standard retrieval instruction prefix on the query side, documents truncated to 2,048 tokens, cosine similarity over the full corpus. Both are run over the same corpora, the same query slice, and the same metric code as the LLM rows.

B Cost Measurements

Measured TFLOPs crossovers are ~15.7k tokens against LocalOnly and ~65.9k against CorpusProbe; latency crossovers are ~6.4k and ~14.0k, an order of magnitude earlier because probes batch while a monolithic read does not. Per-document compute is computed analytically from measured prompt token counts using the judge's geometry (3.633B non-embedding parameters, 36 layers, hidden size 2560), with the quadratic attention term included per call.

The context wall is a property of budget and hardware rather than of the method under test. At the 16,384-token budget used throughout, FullRead fails on 0.7–26.1% of documents depending on domain; a companion run at a smaller budget on 16GB-class hardware failed on 29% overall. We report the budget and hardware with every such number.

C Full Per-Domain Results

Table 3 in the main text reports NDCG@10 and AUC for all eight domains and all six methods. Corpus sizes are 508–1,858 documents per domain (5,650 in total), with FullRead readability ranging from 73.9% to 99.3%. Per-query compute ranges from 898 TFLOPs (FirstP, pony) to 969,097 TFLOPs (CorpusProbe, stackoverflow); the latter is the largest corpus in the benchmark and its documents have a median length an order of magnitude above the smallest domain's.

D Localization Protocol

One annotator marked, for each of 100 documents, the chunks containing evidence that justifies retrieval for the associated query, and where multiple chunks were relevant, which was most relevant. Chunk boundaries match those used by the scoring runs, so annotations and predictions share a coordinate system. Grids are small — a median of 14 chunks with roughly 4 annotated — so we report the chance floor (29.6% for top-1) alongside every accuracy, and the paired McNemar tests in §4.4 compare methods on identical queries rather than comparing marginal means.