
When Does Query Expansion Help Agent Memory?

A Multi-Seed Study of Selective Expansion and Gate Reliability

Salomon Diei¹

Abstract

Long-horizon software agents store execution traces and later retrieve them with natural-language questions, so the query and the memory are written in different “languages”. LLM query expansion can bridge this gap, but it costs generation on every query. We study *Selective Query-Side Expansion* (SQE), which expands a query with hypothetical execution traces and paraphrases only when the top-1 dense retrieval score is low, then fuses all rankings with Reciprocal Rank Fusion. Across 8 resampled memory stores with rebuilt indices (4,000 queries over SWE-smith agent steps), SQE reaches 69.4% Recall@5 while expanding 48% of queries; in a token-measured rerun it uses 55% fewer LLM tokens than always-on expansion. However, its gain over plain dense retrieval is small (+0.9 points, 95% CI [+0.1, +1.7]), Recall@1 drops, and it is statistically indistinguishable from a *random* gate at the same budget. An oracle analysis explains why: expansion helps only 5.3% of queries and hurts 4.2%, so even a perfect gate gains at most 5.3 points. We release all memory stores, per-method summaries, and verification scripts, and argue that confidence gating, not expansion itself, is the open problem.

1. Introduction

Software-engineering agents accumulate long trajectories of observations, commands, file edits, tests, and error traces. These records are valuable only if a later agent can retrieve the relevant fragment when it faces a related problem. Retrieval is hard here because queries and memories are written in different registers: a stored memory may be a traceback or a patch diff, while the later query asks, for example, how an off-by-one line-number bug was fixed.

¹School of Computer Science and Engineering, KOREATECH, Cheonan, Republic of Korea. Correspondence to: Salomon Diei <salomon@koreatech.ac.kr>.

Preprint. October 3, 2026.

Query expansion is a natural remedy. HyDE (Gao et al., 2023) generates a hypothetical document and retrieves with it; for agents, the analogous unit is a hypothetical *execution trace*. Expansion is expensive, though: each expanded query costs several LLM calls and seconds of latency. A practical system should expand only when retrieval is likely to fail. This raises the question we study:

Can a cheap confidence signal decide which agent-memory queries are worth expanding?

We implement SQE (Figure 1), which uses the top-1 dense score as that signal, and evaluate it with an executed random-gate control at matched budget, eight resampled memory stores with rebuilt indices, measured token costs, and an oracle-gate upper bound. Our findings are mostly negative, and we think they are useful precisely because they are negative.

Contributions.

1. We formulate the trace–query mismatch in long-horizon software-agent memory and a selective, query-side-only expansion method that leaves the memory store and indices untouched (Section 3).
2. We run a controlled multi-seed study with an *executed* random-gating baseline at matched expansion rate and provider-measured token costs (Section 5).
3. We show that the top-1-score gate is not better than random, and use an oracle-gate analysis to bound what any gate could achieve in this setting (Section 6).
4. We release seeds, summaries, paired tests, and verification scripts so that every number in this paper can be recomputed (Section 8).

2. Related Work

Dense and sparse retrieval. Dense passage retrieval maps queries and documents to a shared embedding space (Karpukhin et al., 2020); BM25 remains a strong lexical baseline (Robertson & Zaragoza, 2009). Reciprocal Rank Fusion combines ranked lists without training a

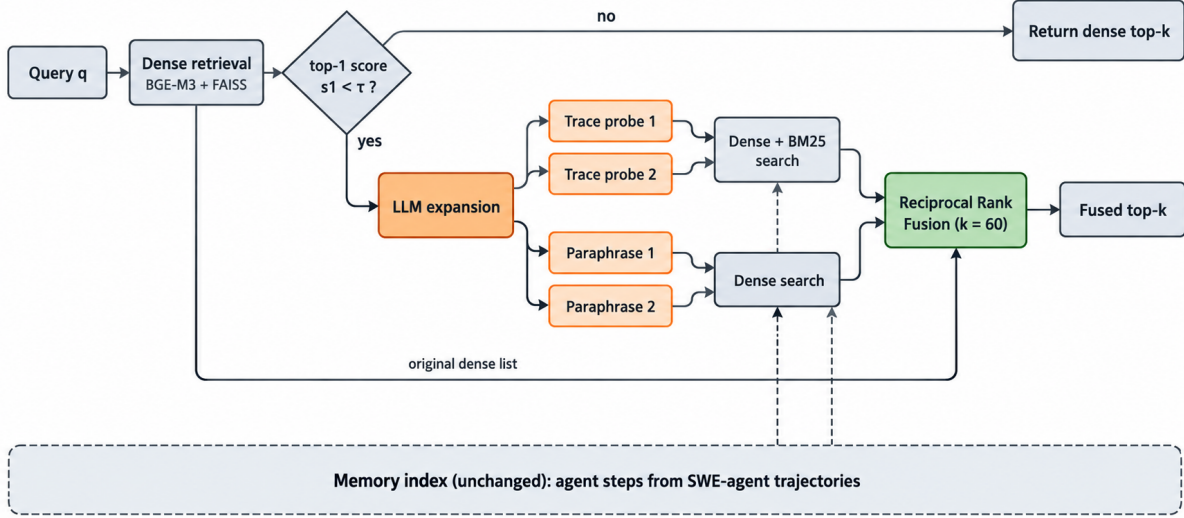


Figure 1. SQE retrieval flow. Dense retrieval runs first; confident queries return the dense list unchanged. Queries whose top-1 score falls below τ are expanded into two hypothetical trace probes (searched with dense and BM25) and two paraphrases (searched with dense only), all against the same unchanged index. The original dense list and the five probe lists are fused with RRF.

ranker (Cormack et al., 2009). We keep both indices fixed and change only the query side.

Query expansion with generated text. HyDE embeds a generated hypothetical document for zero-shot dense retrieval (Gao et al., 2023). EAR expands and reranks candidate queries (Chuang et al., 2023), and HyQE ranks contexts with hypothetical query embeddings (Zhou et al., 2024). Later analysis shows that hypothetical-document gains can partly reflect knowledge leakage from the generator (Yoon et al., 2025). SQE differs in its expansion unit (execution traces) and its explicit generation gate; our results suggest the gate is the weakest component.

Agent memory and software agents. AMA-Bench and MemoryAgentBench evaluate memory over long-horizon and incremental multi-turn interaction (Zhao et al., 2026; Hu et al., 2026). SWE-bench measures whether an agent resolves real issues (Jimenez et al., 2024), and SWE-smith provides large-scale agent trajectories (Yang et al., 2025). Our benchmark is narrower: retrieval of a known target episode, which we treat as an intermediate signal rather than a substitute for task success.

3. Method

Let $M = \{m_i\}_{i=1}^N$ be a store of execution traces. For a query q , dense retrieval returns a ranked list $R_d(q)$ with top-1 cosine score $s_1(q)$. SQE expands q iff $s_1(q) < \tau$; otherwise it returns $R_d(q)$ unchanged. An expanded query yields K hypothetical traces and P paraphrases from an LLM. Each trace is searched with both the dense and the

Algorithm 1 Selective Query-Side Expansion

Input: query q , threshold τ , LLM G , budgets K, P
 $R_d, s_1 \leftarrow \text{DENSE}(q)$
if $s_1 \geq \tau$ **then**
 return R_d
end if
 $\mathcal{T} \leftarrow G_{\text{trace}}(q, K), \mathcal{P} \leftarrow G_{\text{para}}(q, P)$
 $\mathcal{L} \leftarrow \{R_d\} \cup \{\text{DENSE}(p) : p \in \mathcal{P}\}$
 $\mathcal{L} \leftarrow \mathcal{L} \cup \{\text{DENSE}(t), \text{BM25}(t) : t \in \mathcal{T}\}$
return $\text{RRF}(\mathcal{L}, k)$

BM25 index, because traces share exact identifiers with stored steps; each paraphrase is searched with the dense index only. The original list and the $2K + P$ probe lists (each truncated to its top 20) are fused with

$$\text{score}(m) = \sum_j \frac{1}{k + \text{rank}_j(m)}. \quad (1)$$

We use $\tau = 0.65$, $K = P = 2$, and $k = 60$ throughout. The generated traces are retrieval probes, not claims that those executions occurred. Algorithm 1 summarizes the procedure.

4. Experimental Setup

Data. We use the `tool` split of the SWE-smith trajectories (Yang et al., 2025), produced with SWE-agent and Claude 3.7 Sonnet. Each trajectory is cut into single agent steps (thought, action, observation); steps under 100 characters are dropped. A memory store holds 5,000 such steps. For each seed (42–49) we shuffle a shared pool of about

Table 1. Retrieval over 8 resampled memory stores (mean $\pm$ std over seeds, %). SQE expands 48.2% of queries on average; the random gate expands 45.9%.

Method	Seeds	Recall@1	Recall@5	Recall@10
Dense-Only	8	45.5 $\pm$ 3.1	68.5 $\pm$ 2.7	74.2 $\pm$ 2.5
Hybrid-RRF	8	32.5 $\pm$ 1.5	63.5 $\pm$ 2.5	71.8 $\pm$ 2.6
Always-Expand	8	39.8 $\pm$ 3.1	69.2 $\pm$ 3.4	75.9 $\pm$ 2.3
Random-Gated-Expansion	8	42.6 $\pm$ 2.9	68.9 $\pm$ 3.0	75.0 $\pm$ 2.6
Selective-QE	8	44.5 $\pm$ 3.4	69.4 $\pm$ 2.7	75.3 $\pm$ 2.2

Table 2. Paired bootstrap of SQE minus each baseline, Recall@5 points, pooled over 4,000 query rows. $p = 0.000$ denotes $p < 0.001$.

Baseline	Queries	$\Delta R@5$	95% CI	Bootstrap p
Dense-Only	4000	+0.9	[+0.1, +1.7]	0.015
Hybrid-RRF	4000	+5.9	[+4.8, +7.0]	0.000
Always-Expand	4000	+0.1	[-0.4, +0.7]	0.315
Random-Gated-Expansion	4000	+0.5	[-0.2, +1.1]	0.085
Paraphrases-Only	4000	+1.1	[+0.3, +1.9]	0.005
HyDE-Traces-Only	4000	+5.6	[+4.7, +6.6]	0.000

16,500 steps, take 5,000 as the store, rebuild both indices, and sample 500 evaluation targets from the store. The stores are therefore resampled, not disjoint: two stores share about 30% of their steps, and the 4,000 targets contain 3,527 distinct steps. For each target, an LLM that sees the step writes one question meant to identify it, and is told not to copy identifiers, error codes, or variable names. Exactly one stored step counts as relevant per query, so sibling steps from the same trajectory count as misses.

Models. Dense retrieval uses BAAI/bge-m3 with FAISS; sparse retrieval uses BM25. Expansion uses Qwen3.6-35B-A3B served with vLLM.

Baselines. Dense-Only, BM25-Only, Hybrid-RRF (dense + BM25), Paraphrases-Only and HyDE-Traces-Only (always expand with one probe type), Always-Expand (both probe types on every query), and **Random-Gated-Expansion**, which runs the identical expansion code but replaces the gate with seeded random decisions at a comparable rate. The random gate is the key control: it separates the value of *expanding some queries* from the value of *choosing which ones*.

Metrics and inference. We report Recall@{1, 5, 10} and provider-reported tokens and latency. Uncertainty uses paired bootstrap over query rows. These intervals pool queries across seeds and are not corrected for multiple comparisons, so we treat them as descriptive (Section 7).

5. Results

SQE is slightly better than dense retrieval at Recall@5, and worse at Recall@1. Table 1 shows SQE at 69.4% Recall@5 against 68.5% for Dense-Only. The paired differ-

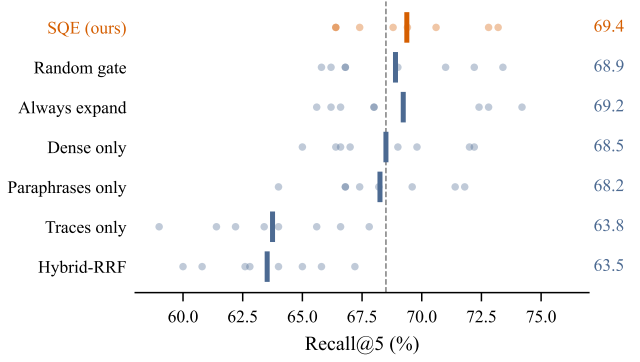


Figure 2. Recall@5 per seed (dots) and 8-seed mean (bar). The dashed line is the Dense-Only mean. Seed variance dominates every expansion effect.

ence is +0.9 points with 95% CI [+0.1, +1.7] (Table 2). At query level, SQE recovers 140 targets that dense retrieval missed but loses 105 that it found, a net gain of 35 out of 4,000 (Table 4, appendix). Recall@1 falls from 45.5% to 44.5%: fusing expanded lists widens the candidate set but disturbs the top position. Figure 2 shows that the seed-to-seed spread (± 2.7 points) is three times larger than the method effect.

The gate does not beat a random gate. Against Random-Gated-Expansion at a similar budget, SQE gains +0.5 points with CI [-0.2, +1.1]. Against Always-Expand the difference is +0.1 [-0.4, +0.7]. Expanding about half of the queries recovers essentially all of the benefit of expanding every query, but *which* half matters little.

Selectivity is still a large cost saving. Figure 3 plots measured tokens (seed 42 rerun) against Recall@5. SQE uses 405 tokens and 2.8 s per query versus 899 tokens and 5.4 s for Always-Expand (Table 5, appendix), with equal Recall@5. Trace-only expansion is the worst operating point: it is expensive and loses 4.7 points to dense retrieval, so hypothetical traces help only when fused with paraphrases and the original query.

6. Why the Gate Fails

Low confidence marks hard queries, not fixable ones. On seed 42, expanded queries reach 61.7% Recall@5 and non-expanded ones 75.9% (Table 6, appendix). The top-1 score does find hard queries, but expansion does not reliably repair them.

The threshold barely matters. Figure 4 sweeps τ on seed 42. Recall@5 stays within a 2.2-point band while the expansion rate goes from 0% to 100%, and the best post-hoc threshold (0.62) beats the fixed one by only 0.8 points. Held-out selection does not change this picture: leave-one-

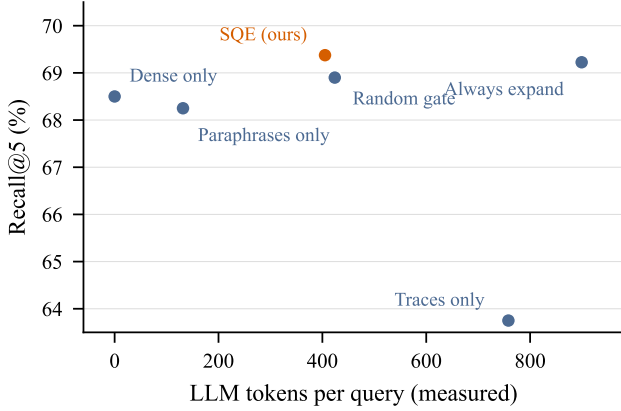


Figure 3. Cost against quality. x : provider-measured LLM tokens per query (seed 42); y : 8-seed mean Recall@5. SQE matches Always-Expand at roughly half the tokens, but so does the random gate.

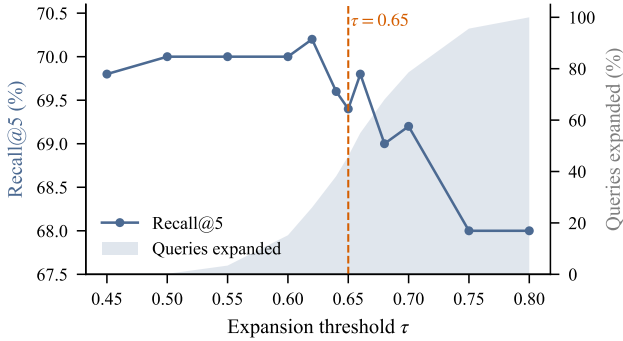


Figure 4. Post-hoc threshold sweep on seed 42. Recall@5 (line, left axis) varies by about 2 points while the fraction of expanded queries (shaded, right axis) spans 0–100%.

seed-out threshold selection gives +0.6 points over dense retrieval with CI $[-0.1, +1.3]$ (Table 7, appendix), and BM25-agreement gate variants do not close the gap (Table 8, appendix).

There is little headroom for any gate. To bound what a better gate could achieve, we combine executed Dense-Only and Always-Expand rows on the held-out half of each seed. An oracle that expands exactly the queries that expansion helps would reach 73.5% Recall@5, against 68.2% for dense retrieval (Figure 5). Expansion helps 5.3% of queries and hurts 4.2%; the remaining 90% have the same outcome either way. With helpful and harmful cases this balanced and this rare, a scalar confidence score has little signal to exploit.

7. Limitations

Queries. Evaluation queries are generated by an LLM that sees the target trace, which can leak target-specific wording and overstate discriminability. A human-audit packet is

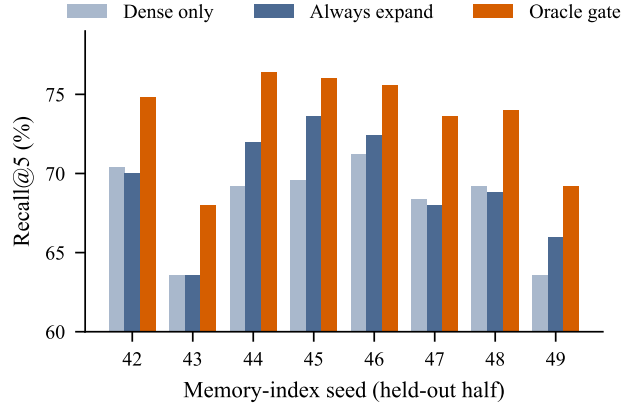


Figure 5. Held-out oracle headroom. Even a perfect per-query gate gains about 5 points over dense retrieval; always expanding gains about 1.

prepared but unlabeled.

Relevance labels. Only the exact source step is relevant. Adjacent steps of the same trajectory can answer the same question, so Recall@ k may understate all methods, and possibly expansion more than dense retrieval.

No downstream outcome. Recall@ k is an intermediate signal; we do not measure whether retrieved memories change agent Pass@1 on SWE-bench-style tasks (Jimenez et al., 2024).

Statistics. Bootstrap intervals pool queries across seeds, but the seeds share about 30% of their memories and some targets, so the 4,000 rows are not independent and the intervals are likely too narrow. No multiplicity correction is applied. A confirmatory study should pre-register one contrast and use seed- or task-clustered inference.

Scope. One embedder, one generator, one trace source, and one memory size. The upstream dataset revision was not pinned for the completed runs, and generator inference seeds were not stored; the released sampled stores are therefore the fixed evidence.

8. Reproducibility

Code, paper sources, and per-seed summaries are at <https://github.com/Salomondiei08/sqe-experiment>; memory stores, evaluation pairs, and result artifacts are at <https://huggingface.co/datasets/TheReinventGuy/sqe-experiment>. `scripts/07_verify_experiment.py` recomputes all metrics from result files and checks that each table matches its declared source. Every figure in this version is produced by `make_figures.py` in the repository

directly from the released JSON, except Figure 1, which is an illustration. The upstream dataset revision was not pinned for the completed runs.

9. Conclusion

Selective query-side expansion halves the generation cost of always-on expansion for agent-memory retrieval with no loss in Recall@5. But in our setting the selection itself is no better than chance, the gain over dense retrieval is under one point, and an oracle bound shows little room for any scalar gate. We conclude that the open problem is predicting *which* queries expansion will fix, and that future work should evaluate learned gates against random-budget controls and measure downstream agent success.

Impact Statement

This work studies retrieval for software-engineering agents. Better memory retrieval could make such agents more capable; we do not foresee negative impacts beyond those of software agents generally. All data are public agent trajectories.

Acknowledgments and Disclosure

An earlier version of this work was presented at NAFSIK 2026. The author used LLM-based assistance to organize the experiment package, write audit scripts, edit text, and draw Figure 1. All numbers come from executed result files, and the author reviewed every claim.

References

- Chuang, Y.-S., Fang, W., Li, S.-W., Yih, W.-t., and Glass, J. Expand, rerank, and retrieve: Query reranking for open-domain question answering. In *Findings of the Association for Computational Linguistics: ACL 2023*, 2023. doi: 10.18653/v1/2023.findings-acl.768. URL <https://aclanthology.org/2023.findings-acl.768/>.
- Cormack, G. V., Clarke, C. L. A., and Buettcher, S. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 758–759, New York, NY, USA, 2009. ACM.
- Gao, L., Ma, X., Lin, J., and Callan, J. Precise zero-shot dense retrieval without relevance labels. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1762–1777, Toronto, Canada, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.99. URL <https://aclanthology.org/2023.acl-long.99/>.
- Hu, Y., Wang, Y., and McAuley, J. Evaluating memory in LLM agents via incremental multi-turn interactions. In *International Conference on Learning Representations*, 2026. URL <https://arxiv.org/abs/2507.05257>.
- Jimenez, C. E., Yang, J., Wettig, A., Yao, S., Pei, K., Press, O., and Narasimhan, K. Swe-bench: Can language models resolve real-world github issues? In *International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=VTF8yNQm66>.
- Karpukhin, V., Oguz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., and Yih, W.-t. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6769–6781, Online, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.550. URL <https://aclanthology.org/2020.emnlp-main.550/>.
- Robertson, S. and Zaragoza, H. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389, 2009. doi: 10.1561/15000000019.
- Yang, J., Lieret, K., Jimenez, C. E., Wettig, A., Khandpur, K., Zhang, Y., Hui, B., Press, O., Schmidt, L., and Yang, D. SWE-smith: Scaling data for software engineering agents. *arXiv preprint arXiv:2504.21798*, 2025. URL <https://arxiv.org/abs/2504.21798>.
- Yoon, Y., Jung, J., Yoon, S., and Park, K. Hypothetical documents or knowledge leakage? rethinking LLM-based query expansion. In *Findings of the Association for Computational Linguistics: ACL 2025*, 2025. doi: 10.18653/v1/2025.findings-acl.980. URL <https://aclanthology.org/2025.findings-acl.980/>.
- Zhao, Y., Yuan, B., Huang, J., Yuan, H., Yu, Z., Xu, H., Hu, L., Shankarampeta, A., Huang, Z., Ni, W., Tian, Y., and Zhao, J. AMA-Bench: Evaluating long-horizon memory for agentic applications. *arXiv preprint arXiv:2602.22769*, 2026. URL <https://arxiv.org/abs/2602.22769>.
- Zhou, W., Zhang, J., Hasson, H., Singh, A., and Li, W. HyQE: Ranking contexts with hypothetical query embeddings. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, 2024. URL <https://aclanthology.org/2024.findings-emnlp.761/>.

A. Additional Results

Table 3. Seed-42 results for all methods (%). This seed carries the per-query rows, case analysis, and token measurements.

Method	Recall@1	Recall@5	Recall@10	MRR	Expansion %
Dense-Only	49.0	69.8	75.4	57.4	–
BM25-Only	17.6	31.4	35.8	23.3	–
Hybrid-RRF	35.4	64.0	73.2	47.4	–
Paraphrases-Only	47.0	68.2	75.0	56.0	100.0
HyDE-Traces-Only	36.2	64.0	70.4	47.9	100.0
Always-Expand	42.2	68.0	75.6	53.2	100.0
Random-Gated-Expansion	46.0	69.0	75.8	55.3	47.0
Selective-QE	47.2	69.4	75.4	56.4	46.0

On seed 42, SQE reaches 69.4% Recall@5 against 69.8% for Dense-Only (−0.4) and 64.0% for Hybrid-RRF (+5.4); Recall@1 is 47.2% against 49.0%. The multi-seed aggregate in Table 1 is the primary evidence.

Table 4. Query-level top-5 wins and losses of SQE against Dense-Only.

Split	N	Both hit	SQE win	SQE loss	Both miss	Net
seed 42	500	334	13	15	138	-2
seed 43	500	314	18	18	150	0
seed 44	500	347	17	13	123	4
seed 45	500	335	18	10	137	8
seed 46	500	350	16	11	123	5
seed 47	500	320	17	15	148	2
seed 48	500	322	22	11	145	11
seed 49	500	313	19	12	156	7
All	4000	2635	140	105	1120	35

Table 5. Provider-reported token usage and latency (seed 42, 500 queries, expansion cache disabled).

Method	Expanded	Calls/q	Prompt tok/q	Total tok/q	Latency s/q
Selective-QE	46.0	1.38	209.2	405.1	2.77
Always-Expand	100.0	2.99	455.5	899.3	5.37
HyDE-Traces-Only	100.0	2.00	367.0	758.2	3.99
Paraphrases-Only	100.0	1.00	88.5	131.6	1.02
Random-Gated-Expansion	47.0	1.41	214.6	423.9	2.40

Table 6. Seed-42 gate split by expansion decision.

Subset	Count	Hits@5	Misses@5	Recall@5
Expanded	230	142	88	61.7
Not expanded	270	205	65	75.9

Table 7. Leave-one-seed-out top-1 threshold selection.

Test seed	Train seeds	Threshold	Gate R@5	Dense R@5	Delta	Expansion
42	43,44,45,46,47,48,49	0.64	69.6	69.8	-0.2	38.2
43	42,44,45,46,47,48,49	0.62	66.6	66.4	+0.2	27.6
44	42,43,45,46,47,48,49	0.62	72.6	72.0	+0.6	28.8
45	42,43,44,46,47,48,49	0.62	69.8	69.0	+0.8	28.8
46	42,43,44,45,47,48,49	0.62	72.0	72.2	-0.2	29.0
47	42,43,44,45,46,48,49	0.64	67.4	67.0	+0.4	42.6
48	42,43,44,45,46,47,49	0.66	68.6	66.6	+2.0	58.8
49	42,43,44,45,46,47,48	0.64	66.4	65.0	+1.4	41.2
All	—	—	69.1	68.5	+0.6	36.9

Table 8. Held-out score-only and BM25-agreement gate variants.

Seed	Policy	Threshold	Test R@5	Dense Test R@5	Expansion
42	score_and_no_bm25_agreement	0.50	70.4	70.4	0.0
43	score_or_no_bm25_agreement	0.65	64.0	63.6	79.6
44	score_only	0.75	72.0	69.2	96.4
45	score_or_no_bm25_agreement	0.70	73.6	69.6	90.4
46	score_only	0.65	72.0	71.2	51.6
47	score_and_no_bm25_agreement	0.45	68.4	68.4	0.0
48	score_and_no_bm25_agreement	0.70	68.8	69.2	55.6
49	score_only	0.60	64.4	63.6	19.6
Mean	—	—	69.2	68.2	49.1

Table 9. Seed-42 examples where SQE and Dense-Only differ at top-5. Illustrative only.

Case	Query ID	Dense rank	SQE rank	Expanded	Query
SQE gain	q_0014	>10	3	yes	What regular expression pattern was used to define the digit group variable?
SQE gain	q_0063	8	3	yes	Did the test script confirm that the parser correctly handles NULL comparisons...
SQE loss	q_0002	1	7	yes	Did the test script confirm that the fix for the FOO value was successful?
SQE loss	q_0031	1	7	yes	What was the output when I ran the simple test script to verify the initial cal...