

RAG Components Are Not Independently Rankable

Vishnu Prasad Vijaya Kumar^{1,2} Iaroslav Neverov¹ Pavel Chizhov^{1,2}
Anton Changalidis¹ Vaiva Kazanaviciute¹ Pierre-Carl Langlais¹
Anastasia Stasenko¹ Ivan P. Yamshchikov^{1,2}

¹PleIAs

²CAIRO, Technical University of Applied Sciences Würzburg-Schweinfurt

Correspondence: vishnu@pleias.fr

Abstract

RAG systems are usually compared as complete pipelines, making it difficult to attribute improvements to individual components. We evaluate chunking, retrieval, fusion, and reranking through a full factorial design with paired component substitutions and equal context budgets. Across five benchmarks, the effect of a chunker depends on the retriever that consumes its output, BM25 fusion changes direction by domain, and reranking is the only intervention we test that remains beneficial across upstream configurations. Claim-level transitions further show that similar net recall can conceal very different changes in visible evidence. These findings show that, for evidence delivery, RAG components cannot be ranked independently, and motivate interaction-aware evaluation instead of winner-take-all pipeline comparisons.

1 Introduction

Retrieval-augmented generation is built from interchangeable parts. A collection is chunked, chunks are retrieved, candidates are optionally reordered by a reranker, and some prefix of the result reaches the generator’s context window. Practitioners choose among these parts using end-to-end benchmark scores.

Such comparisons cannot identify what a component does. A leaderboard reports which complete configuration won, not why—the failure to identify the sources of empirical gains that Lipton and Steinhardt (2019) describe. When semantic chunking beats fixed-size chunking, the chunker may be better, or may merely suit the retriever it was paired with, or the dataset may favour the resulting evidence geometry: different claims with different consequences that no full-pipeline comparison can separate.

The comparison is also unfair. Passages are not a fixed unit of context, so five 512-token chunks expose twice the text of five 256-token chunks, and

a system can score higher simply by supplying more material.

We stop before generation. Answer accuracy is a noisy proxy for evidence delivery: a generator can succeed from parametric knowledge despite poor retrieval, or fail despite sufficient evidence. Adding generators would also add another factor whose own behaviour—such as sensitivity to where evidence appears in the context—would mix with the component effects we want to isolate. Evidence delivery is the stage that chunking, retrieval, and reranking directly control, and a claim that never reaches the context cannot be grounded in it.

Instead we run the complete $4 \times 13 \times 4$ factorial grid at five equal token capacities. Because every combination exists, a component effect is a paired difference: hold the question, dataset, capacity, and every other component fixed, change one component, and average.

Three findings follow, and they do not agree.

Chunking and retrieval are complementary rather than separately rankable. On FinanceBench, semantic chunking relative to Fixed-512 costs 2.6 points under dense retrieval and gains 2.7 under late interaction—an interaction of +5.24 points [2.37, 8.23]. A global chunker ranking averages this away.

Lexical fusion is not a safe default. Adding BM25 to a matched neural parent helps on MultiHop-RAG, QMSum and TAT-QA and harms on FinanceBench and LegalBench-RAG. The pooled effect is +0.91 points with an interval spanning zero, describing none of the five domains.

Reranking is the exception. It stays positive after upstream choices vary: in all 240 dataset $\times$ chunker $\times$ retrieval-family $\times$ reranker cells, and in 776 of 780 cells at the level of individual retrievers.

Our contributions are:

- **A paired-substitution framework.** Token-normalized comparison, claim-level recall,

and paired analysis each exist in prior work; we combine them in one complete $4 \times 13 \times 4$ grid, which yields exact counterfactual pairs for component and interaction effects.

- **Component effects are conditional upstream but stable downstream.** The better chunker depends on the retriever, and the effect of BM25 fusion depends on the dataset; both reverse sign. Reranking is positive in all 240 dataset $\times$ chunker $\times$ retrieval-family $\times$ reranker cells.
- **Claim transitions distinguish interventions with equal net effects.** Separating claims gained from claims lost shows whether a substitution exchanges visible evidence or adds it.

2 Related work

Claim-level RAG evaluation. Automated RAG evaluation frameworks assess dimensions such as context relevance, answer faithfulness, and answer relevance, including RAGAs and ARES (Es et al., 2024; Saad-Falcon et al., 2024). Fine-grained factual evaluation has also decomposed generated text into independently verifiable units: FActScore introduced atomic-fact evaluation, while RefChecker represented claims as claim triplets (Min et al., 2023; Hu et al., 2024). Most directly, RAGChecker defines retriever claim recall as the proportion of reference-answer claims entailed by retrieved chunks, alongside several generator diagnostics (Ru et al., 2024). We adapt this claim-level construct specifically to upstream evidence delivery: reference answers are decomposed into claims and each visible context is checked for claim support, but no answer is generated and no generator-quality metrics are computed.

Context normalization and chunking. Comparisons between retrieval and long context-processing have emphasized the trade-off between answer quality, context length, and computation (Li et al., 2024). Chunk granularity itself has been studied through passages, sentences, and propositions: Dense X evaluates four dense retrievers and limits retrieved context to fixed token budgets, demonstrating that granularity affects both retrieval and downstream QA (Chen et al., 2024). Thus, neither token normalization nor multi-retriever chunking evaluation is new individually. Our distinction is to estimate paired chunker effects across

sparse, dense, late-interaction, fused, and reranked pipelines, revealing when the effect of a chunker changes with retrieval architecture or dataset.

Interaction-aware evaluation. IR evaluation has long recognized that system effects can vary across topics, motivating resampling and interaction-aware statistical analysis (Voorhees et al., 2017). Concurrent preprints apply factorial designs to compound-AI answer quality and to complementarity among document-, query-, and answer-side RAG interventions (Lukassen et al., 2026; Chong, 2026). Our scope is different. To our knowledge, this is the first large-scale factorial audit of upstream RAG evidence delivery that combines equal-token exposure with claim-level semantic recall and paired interventions across chunking, retrieval architecture, lexical fusion, and reranking. This design identifies not only average component effects, but also chunker-retriever compatibility, domain-conditioned sign reversals, and whether an intervention makes new reference claims visible or removes previously visible ones.

3 Controlled Factorial Evaluation of RAG Components

3.1 Benchmarks

We evaluate on five benchmarks spanning financial, legal, multi-document, meeting, and numerical reasoning tasks: FinanceBench (Islam et al., 2023), LegalBench-RAG (Pipitone and Alami, 2024), MultiHop-RAG (Tang and Yang, 2024), QMSum (Zhong et al., 2021), and TAT-QA (Zhu et al., 2021). Table 1 summarizes the task, evaluation subset, document count, and question count for each benchmark.

3.2 Factorial Component Grid

A configuration is $x = (c, r, z)$, comprising a chunker c , first-stage retrieval condition r , and reranking condition z . As summarized in Table 2, we evaluate the complete $4 \times 13 \times 4$ factorial grid rather than selecting a small number of end-to-end pipelines. Every configuration is evaluated on every dataset at five equal-token context capacities.

All retrievers return at most 20 candidates. Rerankers reorder this fixed set and cannot introduce new fragments, whereas fusion may change candidate membership. The cutoff bounds reranking by construction: evidence below rank 20 is invisible to every downstream condition, so our reranker effects are lower bounds on what an unre-

Table 1: Evaluation datasets.

Benchmarks	Task and evaluation subset	Documents	Questions
FinanceBench (Islam et al., 2023)	Financial-report QA; public 150-question set	11,948	150
LegalBench-RAG (Pipitone and Alami, 2024)	Legal retrieval; complete frozen archive	714	6,889
MultiHop-RAG (Tang and Yang, 2024)	Multi-document news QA; non-null queries	609	2,255
QMSum (Zhong et al., 2021)	Query-focused meeting summarization	35	244
TAT-QA (Zhu et al., 2021)	Financial table-and-text QA; gold test set	277	1,660
Total		13,583	11,198

Table 2: RAG components evaluated in the factorial experiment. The pipeline grid contains $4 \times 13 \times 4 = 208$ configurations, each evaluated at five context capacities.

Component	Group	Variants	Levels
Chunking		Fixed-256, Fixed-512, Sentence-512, Semantic-512	4
Retrieval	Sparse	BM25	13
	Dense	EmbeddingGemma-300M, Qwen3-Embedding-0.6B, GTE-ModernBERT, LFM2.5-Embedding-350M	
	Late interaction	GTE-ModernColBERT, LFM2.5-ColBERT-350M	
	Fusion	BM25-RRF with each of the six neural retrievers	
Reranking		None, GTE-Reranker-ModernBERT, Qwen3-Reranker-0.6B, Llama-Nemotron-Rerank-1B-v2	4
Context capacity		256, 512, 1,024, 2,048, 4,096 visible content tokens	5

stricted pool would allow. Context is normalized by tokens rather than by k . Ranked chunks are added until the capacity is reached, and the remaining budget is filled with an exact prefix of the boundary chunk. This prevents differences in chunk length from changing how much text the evaluator sees.

We group first-stage retrieval into three architecture families—sparse, dense, and late interaction—and treat BM25-RRF fusion separately, since fusion is an intervention applied to a neural retriever rather than a distinct architecture. Section 4.3 analyses it as a paired contrast against each matched neural parent; averaging over six parents would conflate architecture with fusion, so it is excluded from the chunker $\times$ family analysis.

3.3 Claim-level outcomes and paired substitutions

We adapt RAGChecker’s atomic-claim evaluation to retrieval only. In **Stage 1**, reference answers and official supports are used once to construct 38,685 question-level reference claims. In **Stage 2**, a pointwise judge receives only the question, reference claims, and one retrieved fragment, returning one Boolean support decision per claim. It does not observe the dataset, pipeline identity, rank, score, capacity, reference answer, or official evidence. We generate no answers and compute no

generator-quality metrics. Across the grid, Stage 2 comprises 5,441,601 fragment-evaluation requests and 18,679,678 claim-level support decisions.

The judge is `google/gemma-4-31B-it`, with reasoning disabled, temperature 0, and a fixed decoding seed. Prompts and schemas are given in Appendix A.

For dataset d and question q , let $G_{d,q}$ be the set of reference claims. A pipeline configuration $x = (c, r, z)$ consists of a chunker c , a first-stage retriever r , and a reranker z . Let $V_{d,q}(x, b)$ be the set of fragments visible under x at a context capacity of b tokens: the complete chunks that fit, plus the prefix of the boundary chunk. Writing $f \models g$ when fragment f supports claim g , our primary outcome is question-level claim recall:

$$Y_{d,q}(x, b) = \frac{1}{|G_{d,q}|} \sum_{g \in G_{d,q}} \mathbf{1}[\exists f \in V_{d,q}(x, b) : f \models g], \quad (1)$$

where $\mathbf{1}[\cdot]$ is 1 if the condition holds and 0 otherwise. A claim is therefore covered when at least one visible fragment supports it.

We do not report fragment precision: the number of visible fragments varies with chunk size at fixed capacity, so precision is not comparable across the chunker substitutions we estimate.

We estimate component effects using paired substitutions. For example, the effect of replacing chunker c with c' is

$$\Delta_{d,q,r,z,b}^{c' \leftarrow c} = Y_{d,q}(c', r, z, b) - Y_{d,q}(c, r, z, b). \quad (2)$$

The question, dataset, retriever, reranker, and capacity are identical on both sides; only the chunker changes. Retrieval, fusion, and reranking effects are defined analogously, and interactions are differences between paired effects: whether the same substitution behaves differently under sparse, dense, or late-interaction retrieval. Every effect is reported against a fixed baseline—Fixed-512, BM25, the matched neural parent, and no reranker—so a positive value always means the named condition delivers more claim coverage.

Questions are weighted equally within each dataset, the five datasets equally in pooled estimates, and capacities and unchanged component levels evenly across the grid. Confidence intervals use 2,000 deterministic paired bootstrap replicates with seed 42, resampling questions within datasets and hierarchically for pooled estimates. Our causal interpretation is local to substitutions within this frozen grid; it implies neither universal component effects nor answer-level correctness.

A blinded human audit of 100 judge-positive and 100 judge-negative decisions confirmed all negatives but only 68 of the positives: the judge rarely misses support but grants it too readily (Appendix A.4).

Complete model and retrieval configurations, including representations, indexing settings, and inference parameters, are provided in Appendix B.

4 Results

The factorial analysis reveals a clear asymmetry between upstream and downstream interventions. Chunking and fusion effects vary across datasets and retrieval regimes, whereas learned reranking remains beneficial after nearly every upstream configuration.

4.1 Chunk Geometry Does Not Determine the Best Chunker

Dataset manifests reveal substantially different evidence geometry. In QMSum, only 4% of official evidence spans are fully contained within a Fixed-256 chunk, compared with 17% for Fixed-512 and 19% for Sentence-512. A reference span consequently touches 5.57 Fixed-256 chunks on average,

compared with 3.27 Fixed-512 chunks. Evidence in MultiHop-RAG and TAT-QA, by contrast, is usually contained within a single chunk.

These intrinsic differences do not produce a consistent chunker ranking. Figure 1 reports paired changes in claim recall for each chunker substitution relative to Fixed-512. The direction and magnitude of these effects vary across datasets, and no alternative dominates universally. QMSum makes the disconnect concrete: it has the worst Fixed-256 containment of any dataset, yet Fixed-256 improves claim recall there under dense retrieval (Section 4.2). Geometry describes the representation given to retrieval; whether it helps depends on the retriever that consumes it.

4.2 Chunking and Retrieval Are Complementary

The effect of a chunking choice depends on the retriever that consumes its output. Figure 2 compares each chunker with Fixed-512 under sparse, dense, and late-interaction retrieval, with reranking disabled. Effects are averaged across the five capacities and equally across models within each family.

On FinanceBench, Semantic-512 changes claim recall by +0.27 percentage points under sparse retrieval, −2.59 under dense, and +2.65 under late interaction, a late-interaction-minus-dense interaction of +5.24 points [2.37, 8.23].

The pattern also holds within matched backbones, so it does not come from averaging four dense models against two late-interaction ones: the effect rises from −3.1 to +1.4 for GTE and from +1.6 to +3.9 for LFM2.5 when moving from dense to late interaction (Figure 7).

Reversals occur elsewhere too. On LegalBench-RAG, Fixed-256 changes recall by −2.17 under sparse but +0.75 under dense, an interaction of +2.92 [2.50, 3.30]; on QMSum the same substitution gives −1.63 and +2.01, an interaction of +3.64 [1.47, 5.86].

MultiHop-RAG bounds the claim: Fixed-256 helps under sparse (+5.34), dense (+4.34), and late interaction (+4.64). Its evidence is short and almost always sits inside one chunk, so smaller chunks concentrate it rather than splitting it, and every family benefits. Component effects are conditional, not arbitrary: where evidence geometry is uniform across a corpus, a chunker choice can transport.

Overall, chunker effects change direction across

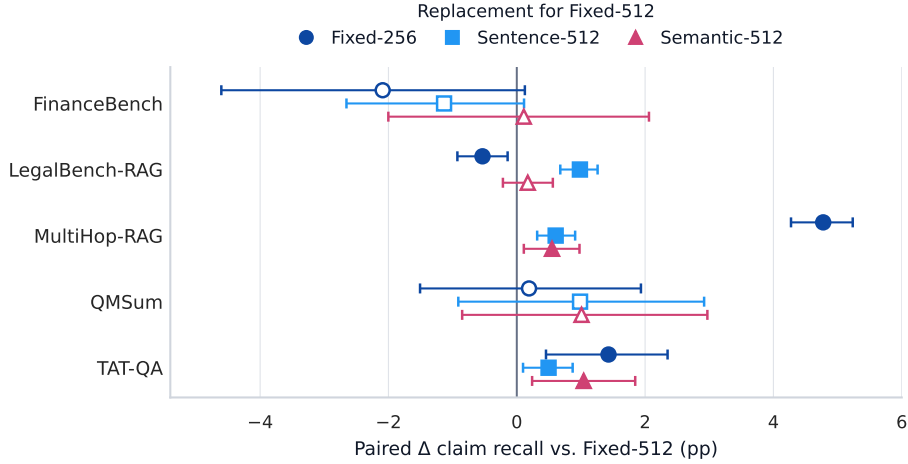


Figure 1: **The effect of replacing Fixed-512 varies across datasets.** Points show paired changes in claim recall, averaged equally across the sparse, dense, and late-interaction retrieval families and across context capacities, with reranking disabled. BM25–RRF fusion conditions are excluded (Section 3.2). Error bars denote 95% paired-bootstrap confidence intervals; filled markers indicate intervals that exclude zero.

retriever families in four of the five datasets. This means that a chunker should not be judged on its own: whether it helps depends on the retriever used with it.

4.3 BM25 Fusion Is Conditional on the Upstream Pipeline

Figure 3 compares each BM25–RRF condition with its exact neural parent while holding the dataset, chunker, context capacity, and no-reranker condition fixed. Averaged over the six matched neural parents and four chunkers, fusion exhibits a dataset-level sign reversal: claim recall decreases by 5.92 percentage points on FinanceBench and 2.85 points on LegalBench-RAG, but increases by 4.53 points on MultiHop-RAG, 2.46 points on QMSum, and 6.33 points on TAT-QA.

The reversal is not driven by one parent model. On FinanceBench and LegalBench-RAG, none of the six neural retrievers improves after fusion under any chunker; on MultiHop-RAG, all six improve under every chunker, with five or six on QMSum and four or five on TAT-QA. The same matched intervention reliably harms two datasets and helps three. Magnitude is less even: on TAT-QA, fusion with LFM Dense gains 18–21 points, several times any other parent, so that dataset’s +6.33 mean rests largely on one model pair (Appendix C).

Averaging these opposing effects produces a pooled gain of only 0.91 points, with a 95% confidence interval spanning zero [−3.09, 4.82]. This

pooled estimate conceals a qualitative interaction: BM25 fusion changes not only in magnitude but also in direction across datasets. Fusion should therefore be treated as a domain-conditioned intervention rather than a generally beneficial retrieval enhancement.

4.4 Net Effects Conceal Claim-Level Transitions

4.5 Reranking Is Robust to Upstream Pipeline Choices

Unlike chunking and fusion, reranking produces positive effects in nearly every tested configuration. Figure 4 compares each learned reranker with no reranker, holding the dataset, question, chunker, first-stage retriever, and capacity fixed. Averaged equally across datasets and retrieval families, GTE-Reranker-ModernBERT improves claim recall by +5.74 percentage points [4.10, 7.87], Qwen3-Reranker-0.6B by +6.50 [4.73, 7.68], and Llama-Nemotron-Rerank-1B-v2 by +8.76 [6.83, 10.51].

The consistency holds below the pooled average. All 240 dataset $\times$ chunker $\times$ retrieval-family $\times$ reranker comparisons are positive (here fusion counts as a fourth upstream group, since reranking is applied after it), as are 776 of 780 cells at the level of individual retrievers; the four exceptions are small and localized. The Nemotron reranker is positive on every dataset, from +5.28 on QMSum to +11.05 on FinanceBench, and varies with capacity +7.46 at 256 tokens, +11.30

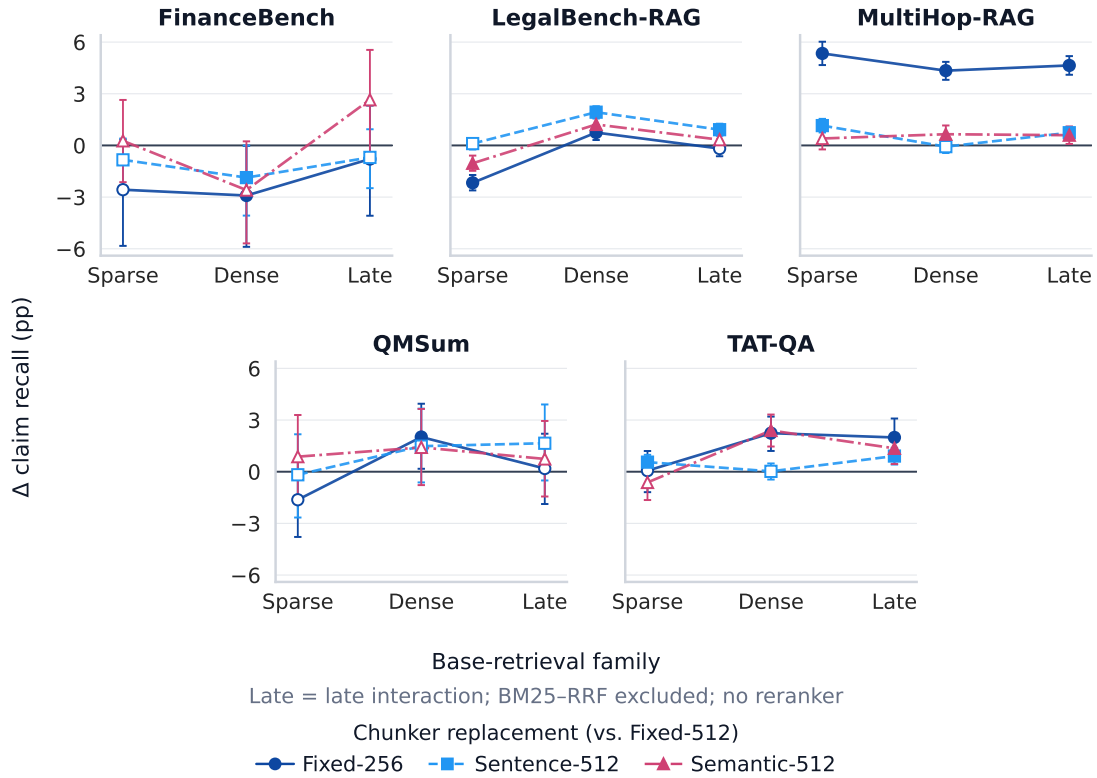


Figure 2: **Chunker effects depend on retrieval family and dataset.** Each panel shows one dataset. Lines represent chunker replacements relative to Fixed-512, while the horizontal axis varies the retrieval family. Points show paired changes in claim recall, averaged across context capacities and equally across models within each family, with reranking disabled. Error bars denote 95% paired-bootstrap confidence intervals; filled markers indicate intervals excluding zero.

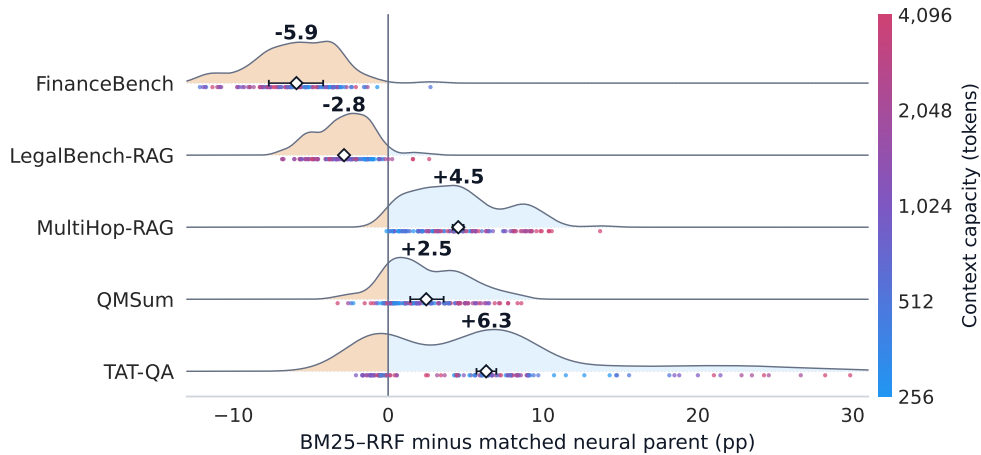


Figure 3: **BM25 fusion has dataset- and pipeline-dependent effects.** Each rug mark represents one fixed chunker, neural parent, and context capacity. Ridges summarize paired changes in claim recall across the tested conditions. Diamonds and lines show dataset-level mean effects and paired-bootstrap 95% confidence intervals.

at 1,024, +5.15 at 4,096—without reversing. The fixed pool explains the shape: at 256 tokens few chunks fit whatever their order, at 4,096 much of the top 20 is visible whatever the ranking, so re-ordering matters most in between.

Reranking differs structurally from the preceding interventions: it reorders a fixed pool rather than changing the chunk representation or adding a retrieval source. Under a constrained budget it can lift evidence above the visibility threshold without

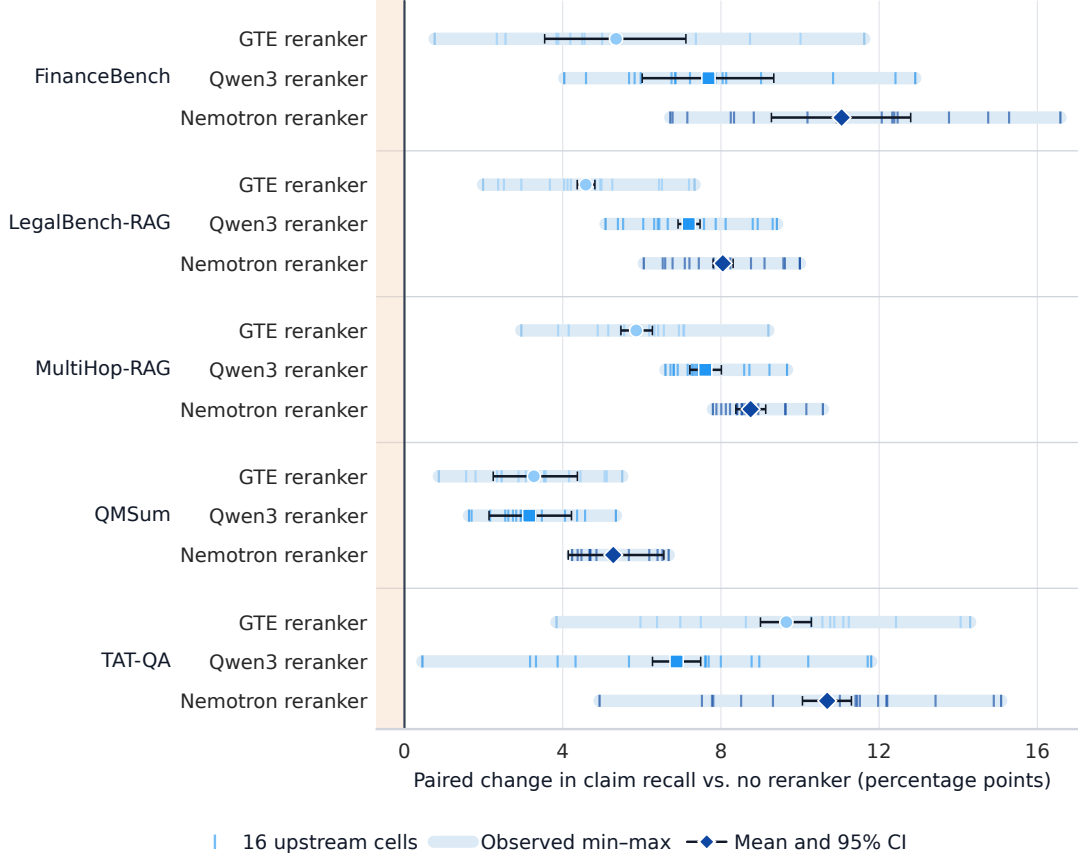


Figure 4: **Learned reranking improves claim recall across upstream configurations and datasets.** Each row compares one reranker with no reranker. Pale segments show the observed minimum–maximum effect across 16 upstream configurations (four chunkers $\times$ four upstream groups: sparse, dense, late interaction, and fusion), and small ticks mark the individual configurations. Colored markers show dataset-level paired means; black whiskers denote 95% paired-bootstrap confidence intervals.

depending on a particular chunker–retriever pairing. Within our frozen grid it is therefore more stable than chunking or fusion—a conclusion local to the evaluated pools, rerankers, datasets, and capacities, not a claim that every reranker helps. Claim recall also ignores order within the visible context: a claim counts as covered wherever it appears. Reranking changes where evidence sits, and generators are sensitive to position (Liu et al., 2024), so a gain in delivered evidence need not become an equal gain in answer quality.

Aggregate recall changes do not reveal which claims change coverage state. We therefore decompose each paired intervention into two mutually exclusive transitions: claims that become covered under the intervention and claims that cease to be covered. Their difference recovers the net recall effect:

$$\Delta\text{Recall} = P(0 \rightarrow 1) - P(1 \rightarrow 0). \quad (3)$$

Figure 5a reports chunker transitions under the complete-grid estimand, averaging across rerankers as well as retrievers and capacities; the chunker effects in Section 4.2 instead condition on the no-reranker setting, so the two are not numerically identical. Figure 5b conditions reranker transitions on the upstream retrieval family, since a reranker is defined only relative to a first-stage ranking. All estimates weight capacities and datasets equally.

Replacing Fixed-512 with Fixed-256 illustrates why a near-zero average effect need not imply that the intervention is inconsequential. Fixed-256 newly covers 7.95% of claims but no longer covers 8.12%, a net change of only -0.16 percentage points (95% CI $[-3.10, 2.64]$; computed from unrounded shares). Thus, approximately 16.1% of claims change coverage state even though aggregate recall changes little. The substitution changes which evidence is visible rather than uniformly improving or degrading coverage.

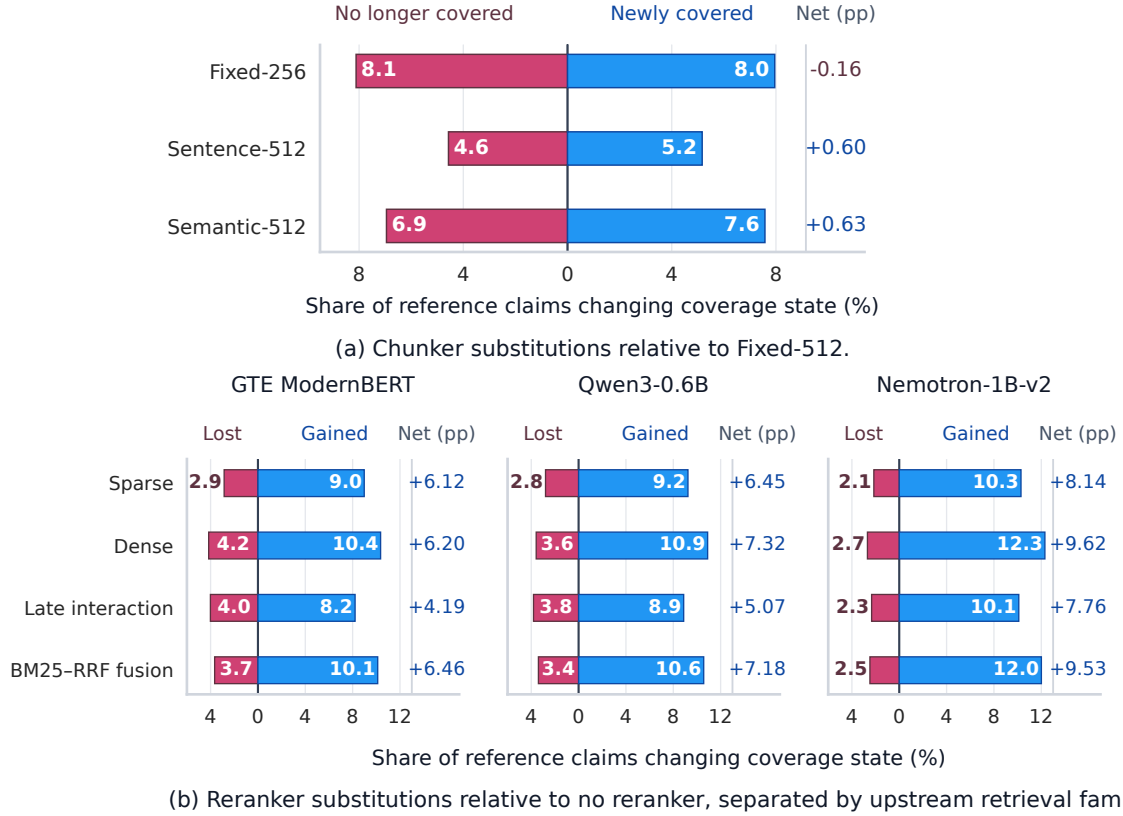


Figure 5: **Component substitutions change which reference claims are covered.** Blue bars show newly covered claims and pink bars show claims no longer covered. Net changes are reported in percentage points. Chunker effects are averaged across datasets, capacities, retrievers, and rerankers. Reranker effects are averaged across datasets, capacities, chunkers, and the models within each retrieval family.

Reranking produces a qualitatively different pattern. Across every reranker and retrieval family, newly covered claims outnumber claims no longer covered. The Nemotron reranker yields the largest net improvement in all four families: +8.14 points with sparse retrieval, +9.62 with dense retrieval, +7.76 with late interaction, and +9.53 with BM25-RRF fusion. Its newly covered share ranges from 10.10% to 12.30%, whereas only 2.15%–2.69% of claims cease to be covered. The GTE and Qwen3 rerankers are also repair-dominant, although their gains are consistently smaller. For all three rerankers, the smallest net improvement occurs after late-interaction retrieval, suggesting that stronger first-stage retrieval leaves less uncovered evidence for reranking to recover.

These results show that net recall can conceal materially different effect compositions. Chunking may exchange one set of visible claims for another, whereas reranking predominantly adds coverage in

this grid. Transition analysis describes how coverage changes under a controlled substitution; it does not by itself identify the mechanism producing that change.

5 Discussion

5.1 Component Quality Is Relational

The pattern in our results is not uniform variance. Chunking and fusion change which candidates exist; reranking only reorders a fixed pool. The interventions that change the representation are the ones whose sign depends on what consumes it, while the one that leaves the candidate set alone transports across upstream regimes. A component’s transportability tracks how much of the pipeline it changes. This qualifies our third finding: reranking may be more stable because it is a smaller intervention, not because reordering is inherently more reliable than representation.

5.2 Implications for Evaluation

Three evaluation practices follow for studies of evidence delivery. Report chunker results together with the retriever they were measured against, since a chunker effect without its retrieval family cannot be interpreted. Compare fusion against its matched neural parent on each target domain; our pooled estimate of +0.91 points describes none of the five datasets. And report sign-consistency counts alongside pooled means, which otherwise cannot distinguish a small effect from opposing large ones.

These practices concern how evidence delivery is measured. Whether the same component choices improve generated answers is a separate question our design does not test. What our results do show is that a leaderboard cannot say which component to change next: the answer depends on the dataset and on what sits either side of the component.

6 Conclusion

We presented a controlled factorial evaluation of RAG evidence delivery across chunking, retrieval, fusion, reranking, and context capacity. Rather than ranking complete pipelines, we estimated paired component substitutions while holding the remaining pipeline fixed and controlling visible context by token capacity.

The results show that component effects are conditional. The effect of a chunker depends on the retriever that consumes its output, and BM25+RRF fusion helps on some datasets while harming others. Reranking improves claim recall in nearly every configuration we test, with smaller gains after late-interaction retrieval. Claim-transition analysis further shows that similar net effects can hide very different changes: chunking often exchanges which claims are covered, whereas reranking mostly adds coverage.

For evidence delivery, these findings argue against treating RAG components as independently rankable modules. Evaluation should combine equal-context comparison, paired substitutions, interaction analysis, and claim-level diagnostics, so that it shows not only which pipeline performs best but which components drive the difference and under what conditions.

Limitations

Our conclusions are local to the five datasets, component implementations, candidate pools, context capacities, and retrieval evaluator used in this study.

The factorial design identifies controlled effects within this frozen grid, but it does not establish that the same effects will hold for other domains, models, indexing strategies, or context windows. Candidate pools are capped at the first-stage top 20 (Section 3.2), so we cannot measure the ceiling an unrestricted pool would permit.

We evaluate evidence delivery rather than generated answers. Claim recall indicates whether retrieved fragments contain reference claims, but it does not measure answer correctness, faithfulness, fluency, or downstream reasoning. Delivering a claim is necessary for grounding an answer in it, not sufficient: generators differ in how well they use the evidence they receive. Testing whether the component effects reported here carry over to answer quality, using several generators on a subset of the grid, is the most direct next step.

The same frozen model constructs the reference claims and judges support, so the judge may more readily confirm claims phrased the way it would phrase them. A blinded audit confirmed 68 of 100 judge-positive and all 100 judge-negative decisions (Appendix A.4), so the judge is permissive and absolute claim recall is overstated. Paired effects are affected only if the false-positive rate differs between compared conditions—for example, if reranked contexts expose the judge to more fragments that are on-topic but do not support the claim. A larger audit stratified by chunker, reranker, and claim type would bound this more tightly.

Support is judged one fragment at a time, which interacts with chunk size in two opposite ways. Evidence split across fragments cannot be combined, so a claim whose support crosses a chunk boundary is undercounted, as can occur in MultiHop-RAG and TAT-QA; this penalizes finer chunking. At the same time, smaller chunks yield more fragments within a fixed capacity, each another chance for a false positive; this favors finer chunking. Because the number of fragments depends only on the chunker and capacity, this effect is the same on both sides of every reranking and fusion contrast and cancels in the chunker $\times$ retrieval-family interactions. It can bias chunker main effects, such as Fixed-256 against Fixed-512, in a direction the audit cannot determine.

Retrieval implementations are not symmetric. Late-interaction candidates come from approximate FastPLAID search with 4-bit residual compression, whereas dense retrieval is exact and exhaustive, so part of any dense versus late-

interaction difference may reflect implementation rather than architecture. The families are also unbalanced: four dense models against two late-interaction models. Matched GTE and LFM2.5 backbones show the same direction (Section 4.2), which addresses model composition but not search approximation; repeating the comparison with exact MaxSim retrieval would separate the two.

Finally, equal-token capacity controls context exposure but does not capture all deployment costs. Latency, index size, retrieval throughput, reranking cost, and generator sensitivity to evidence ordering remain outside the present analysis.

References

- Tong Chen, Hongwei Wang, Sihao Chen, Wenhao Yu, Kaixin Ma, Xinran Zhao, Hongming Zhang, and Dong Yu. 2024. [Dense X retrieval: What retrieval granularity should we use?](#) In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 15159–15177, Miami, Florida, USA. Association for Computational Linguistics.
- Ng ST Chong. 2026. Three sides of retrieval: Factorial evidence for document-side, query-side, and answer-side complementarity in RAG. *arXiv preprint arXiv:2607.24781*.
- Shahul Es, Jithin James, Luis Espinosa Anke, and Steven Schockaert. 2024. [RAGAs: Automated evaluation of retrieval augmented generation](#). In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 150–158, St. Julians, Malta. Association for Computational Linguistics.
- Xiangkun Hu, Dongyu Ru, Lin Qiu, Qipeng Guo, Tianhang Zhang, Yang Xu, Yun Luo, Pengfei Liu, Yue Zhang, and Zheng Zhang. 2024. [Knowledge-centric hallucination detection](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 6953–6975, Miami, Florida, USA. Association for Computational Linguistics.
- Pranab Islam, Anand Kannappan, Douwe Kiela, Rebecca Qian, Nino Scherrer, and Bertie Vidgen. 2023. FinanceBench: A new benchmark for financial question answering. *arXiv preprint arXiv:2311.11944*.
- Zhuowan Li, Cheng Li, Mingyang Zhang, Qiaozhu Mei, and Michael Bendersky. 2024. [Retrieval augmented generation or long-context LLMs? a comprehensive study and hybrid approach](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 881–893, Miami, Florida, US. Association for Computational Linguistics.
- Zachary C Lipton and Jacob Steinhardt. 2019. Research for practice: troubling trends in machine-learning scholarship. *Communications of the ACM*, 62(6):45–53.
- Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. [Lost in the middle: How language models use long contexts](#). *Transactions of the Association for Computational Linguistics*, 12:157–173.
- Fabian Lukassen, Christoph Weisser, Thomas Kneib, and Alexander Silbersdorff. 2026. CAFE: A Compound-AI factorial evaluation framework. *arXiv preprint arXiv:2607.10380*.
- Sewon Min, Kalpesh Krishna, Xinxu Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. [FactScore: Fine-grained atomic evaluation of factual precision in long form text generation](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100, Singapore. Association for Computational Linguistics.
- Nicholas Pipitone and Ghita Houir Alami. 2024. LegalBench-RAG: A benchmark for retrieval-augmented generation in the legal domain. *arXiv preprint arXiv:2408.10343*.
- Dongyu Ru, Lin Qiu, Xiangkun Hu, Tianhang Zhang, Peng Shi, Shuaichen Chang, Jiayang Cheng, Cunxiang Wang, Shichao Sun, Huanyu Li, Zizhao Zhang, Binjie Wang, Jiarong Jiang, Tong He, Zhiguo Wang, Pengfei Liu, Yue Zhang, and Zheng Zhang. 2024. RAGChecker: A fine-grained framework for diagnosing retrieval-augmented generation. In *Advances in Neural Information Processing Systems 37 (Datasets and Benchmarks Track)*, volume 37, pages 21999–22027.
- Jon Saad-Falcon, Omar Khattab, Christopher Potts, and Matei Zaharia. 2024. [ARES: An automated evaluation framework for retrieval-augmented generation systems](#). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 338–354, Mexico City, Mexico. Association for Computational Linguistics.
- Yixuan Tang and Yi Yang. 2024. MultiHop-RAG: Benchmarking retrieval-augmented generation for multi-hop queries. *arXiv preprint arXiv:2401.15391*.
- Ellen M. Voorhees, Daniel V. Samarov, and Ian Soboroff. 2017. [Using replicates in information retrieval evaluation](#). *ACM Transactions on Information Systems*, 36(2):12:1–12:21.
- Ming Zhong, Da Yin, Tao Yu, Ahmad Zaidi, Mutethia Mutuma, Rahul Jha, Ahmed Hassan Awadallah, Asli Celikyilmaz, Yang Liu, Xipeng Qiu, and Dragomir Radev. 2021. [QMSum: A new benchmark for query-based multi-domain meeting summarization](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages

5905–5921, Online. Association for Computational Linguistics.

Fengbin Zhu, Wenqiang Lei, Youcheng Huang, Chao Wang, Shuo Zhang, Jiancheng Lv, Fuli Feng, and Tat-Seng Chua. 2021. [TAT-QA: A question answering benchmark on a hybrid of tabular and textual content in finance](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3277–3287, Online. Association for Computational Linguistics.

A Evaluation Prompts, Schemas, and Validation

This appendix reports the complete instructions used for reference-claim construction and point-wise claim-support evaluation. Gemma 4 31B-IT was used as reported in Section 3.3. Thinking was disabled, temperature was 0, top- p was 1, and the random seed was 42. Structured outputs were generated with strict JSON schemas using the guidance backend. The static prompt text below is reproduced verbatim; expressions enclosed in angle brackets denote values inserted separately for each question or retrieved fragment.

A.1 Reference-Claim Construction

The first evaluation stage converted each benchmark reference answer into a set of standalone reference claims. Official benchmark evidence was supplied to ground the claims and retain provenance.

Stage 1: Claim-construction system message

You create reference claims for evaluating retrieved evidence. The user provides a question, reference answer, claim-coverage requirement, optional answer annotations, and labelled official evidence. Create a complete, non-redundant set of standalone atomic claims that follows the claim-coverage requirement.

Coverage comes before minimization. Include every material assertion, explanation, qualifier, comparison, and calculation required by the specified coverage. Remove a claim only when it repeats the same proposition and removing it does not reduce the required coverage.

Each claim must state one independently verifiable proposition and be understandable without the question, answer, other claims, or evidence. Resolve pronouns and preserve every necessary entity, relationship, date, unit, scope, qualifier, definition, and comparison direction. Keep an exception or limitation in the same claim as the proposition it qualifies; never separate their meanings.

For a prose answer, decompose every material assertion and explanation. For a short entity or scalar answer, combine the question and answer into standalone propositions; never write "the answer is X." For a

calculated answer, create claims for the required operand facts and one final calculation claim containing the operation, operands, result, and unit or scale. If that result directly establishes a yes/no or comparison conclusion, include it in that calculation claim instead of repeating the result.

Use official evidence as grounding and provenance for claims needed to establish the reference answer. Do not create a claim merely to cite every supplied item: evidence may contain headings, bridge text, redundant context, adjacent rows, or distractors. Leave an item uncited when it adds no material proposition required by the answer. Ignore unrelated details, rebuttals, or counterarguments in the same item. In answer-first mode, evidence does not introduce extra claims.

Every claim must cite all and only the official evidence items that support it. evidence_ids may contain only supplied labels such as E001. Evidence IDs represent provenance, not dependencies between claims. Never put claim IDs in evidence_ids, evidence labels in claim text, or use outside knowledge. Before returning, verify completeness, atomicity, grounding, arithmetic, units, and non-redundancy. Remove any claim whose deletion would not reduce required answer or evidence coverage. Return only the required structure.

The corresponding user message was rendered from the following template.

Stage 1: User-message template

```
QUESTION:
<question>

REFERENCE ANSWER:
<reference answer>

<optional reference-answer annotations>

CLAIM COVERAGE REQUIREMENT:
<coverage requirement>

ALLOWED OFFICIAL EVIDENCE IDS:
<E001, E002, ...>

OFFICIAL EVIDENCE:

[E001]
<official evidence text>

[E002]
<official evidence text>

...
```

When benchmark annotations were available, they were inserted using the following fields. Only available fields were included.

Optional reference-answer annotations

```
REFERENCE ANSWER ANNOTATIONS:
Answer type: <answer type>
Answer unit or scale: <scale>
Official answer rationale:
<rationale>
Official derivation: <derivation>
Official facts:
- <fact>
...
```

Coverage policies. FinanceBench, LegalBench-RAG, and QMSum used the following answer-first coverage requirement.

Answer-first coverage requirement

Represent every material assertion and explanation in the reference answer. Use official evidence and answer annotations to ground and validate those claims. Do not create claims from redundant, structural, incidental, or unrelated text.

MultiHop-RAG and TAT-QA used the following evidence-fallback requirement. This policy was used when short reference answers did not themselves express all supporting operands or evidence requirements.

Evidence-fallback coverage requirement

The reference answer alone does not contain the complete evidence requirements. Use the question, answer annotations, and official evidence to construct them. Represent every nonredundant evidence fact required to establish the answer.

A.1.1 Claim Output Schema

Claim responses were generated under a strict JSON schema. At request time, the enum under `evidence_ids` was replaced by the exact set of official evidence labels supplied for the question.

Stage 1: Claim-construction schema

```
{
  "type": "object",
  "additionalProperties": false,
  "required": ["claims"],
  "properties": {
    "claims": {
      "type": "array",
      "minItems": 1,
      "items": {
        "type": "object",
        "additionalProperties": false,
        "required": [
          "claim_id",
          "text",
          "evidence_ids"
        ],
        "properties": {
          "claim_id": {
            "type": "string"
          },
          "text": {
            "type": "string"
          },
          "evidence_ids": {
            "type": "array",
            "minItems": 1,
            "items": {
              "type": "string",
              "enum": [
                "E001",
                "E002",
                "...",
              ]
            }
          }
        }
      }
    }
  }
}
```

```
}
}
}
}
```

A.2 Pointwise Claim-Support Evaluation

The second stage evaluated whether each retrieved fragment supported each reference claim. Every request contained one retrieved fragment and all reference claims for the corresponding question. Decisions were therefore pointwise with respect to a single fragment: evidence from different fragments could not be combined.

Stage 2: Pointwise-support system message

Decide whether the single retrieved chunk supports each reference claim. A claim is supported only when this chunk by itself entails the complete claim. Do not use outside knowledge, the question alone, or information from another chunk. A partial match is not enough.

For a calculated quantitative claim, mark it supported when the chunk contains every required input with the labels and units needed to interpret it, and the stated calculation follows correctly from those inputs; the derived result does not need to appear verbatim in the chunk. When the claim explicitly states a formula or operational definition, evaluate that stated formula or definition; do not replace it with a different conventional definition. Return exactly one decision for every reference claim in the required structure.

The corresponding user message was rendered as follows.

Stage 2: User-message template

```
QUESTION:
<question>

REFERENCE CLAIMS:
[CL001] <claim text>
[CL002] <claim text>
...

RETRIEVED CHUNK:
[CO01] <retrieved fragment>
```

A `supported=true` decision therefore means that the single displayed fragment entails the complete target claim by itself. A fragment that supports another claim, or only part of the target claim, receives `supported=false` for that target claim.

A.2.1 Support Output Schema

For a request containing N reference claims, both `minItems` and `maxItems` were set to N . The `claim_id` enumeration was also replaced by the exact claim identifiers supplied in that request.

Stage 2: Pointwise-support schema

```
{
  "type": "object",
  "additionalProperties": false,
  "required": ["claim_results"],
  "properties": {
    "claim_results": {
      "type": "array",
      "minItems": <N>,
      "maxItems": <N>,
      "items": {
        "type": "object",
        "additionalProperties": false,
        "required": [
          "claim_id",
          "supported"
        ],
        "properties": {
          "claim_id": {
            "type": "string",
            "enum": [
              "CL001",
              "CL002",
              "... "
            ]
          },
          "supported": {
            "type": "boolean"
          }
        }
      }
    }
  }
}
```

A.3 Validation and Retry Policy

A.3.1 Claim Validation

Claim responses were accepted only when they completed before the 8,192-token output ceiling, decoded as a JSON object containing at least one claim, and passed the following protocol checks:

- every claim was represented by a JSON object;
- claim text was non-empty and not duplicated within the response;
- evidence labels such as [E001] did not appear inside claim text;
- every claim cited at least one official evidence ID;
- evidence IDs were strings and were unique within each claim; and
- every cited evidence ID occurred in the allowed set supplied with the question.

After validation, claim identifiers were assigned canonically as CL001, CL002, ... in output order. The original model-generated identifier did not determine the final claim identifier.

The validator did not require every supplied official evidence item to be cited. Official evidence could contain headings, bridge text, adjacent rows, redundant context, or distractors. Uncited supplied evidence IDs were retained in an audit field but did not invalidate an otherwise valid claim set.

These checks validate response structure and evidence provenance; they do not independently establish semantic entailment between a claim and its cited official evidence. We deliberately used no lexical-overlap, number-matching, OCR-normalization, or arithmetic heuristics to accept or reject claims. Semantic grounding, arithmetic, units, completeness, and atomicity were governed by the frozen prompt and inspected through manual auditing.

A.3.2 Claim Retries

A failed initial claim response received at most three retries, producing a maximum of four attempts. Every retry reused the complete original request and appended the following instruction.

Stage 1: Retry instruction

Recreate the complete claim set from the original input and apply every system rule. Ensure the claims jointly establish the reference answer, every claim is necessary and grounded, calculations and units are exact, and evidence_ids contain only supplied official evidence labels—not claim IDs. Return the complete corrected structure.

For every attempt, we retained the exact request body, raw response, proposed claim objects, failure code, evidence IDs, token usage, finish reason, and latency. Failed-attempt claims were never passed to the support-evaluation stage.

Following audited retry exhaustion, two responses remained invalid solely because they repeated byte-identical claim text. These duplicates were losslessly consolidated: one copy of each distinct text was retained and its evidence-ID list was replaced by the union of the duplicated copies. No claim text was rewritten and no evidence ID was invented. This removed three duplicate claim objects across two questions.

A.3.3 Support Validation

Support responses were accepted only when they:

- completed before the 2,048-token output ceiling;
- decoded as a valid JSON object;

- returned a Boolean decision for every supplied claim ID;
- returned each supplied claim ID exactly once; and
- contained no unknown or duplicated claim IDs.

A failed support response received one retry, for a maximum of two attempts. The retry reused the original request and appended the following instruction.

Stage 2: Retry instruction

Check the result again. Return exactly one decision for every supplied claim. Mark supported as true only when the single retrieved chunk entails the complete claim by itself.

No support request required a retry in the final production run.

A.4 Human Audit of Support Decisions

The Stage 2 judge marked 539,201 of 18,679,678 claim–fragment decisions as supported (2.9%). We audited a blinded sample of 100 judge-positive and 100 judge-negative decisions. On this balanced sample, overall agreement was 84.0% ($\kappa = 0.68$): the annotator confirmed 68 of 100 positive decisions (95% Wilson interval 58.3–76.3%) and all 100 negative decisions (96.3–100.0%). Reweighting the same cases to the production frequencies of dataset, chunker, and judge label gives 68.6% agreement on positives, 100.0% on negatives, and 99.1% overall ($\kappa = 0.81$); the intervals above describe the sampled cases, not the reweighted rates. The judge therefore rarely misses support but grants it too readily, so absolute claim recall is overstated. Because this bias is present on both sides of every paired substitution, it affects effect estimates only where the false-positive rate differs between the compared conditions. Table 3 reports the breakdown by dataset.

A.5 Inference Parameters

Claim construction used a 22,000-token model context, an 8,192-token output ceiling, tensor parallelism of two, and at most three retries after the initial attempt. Pointwise support evaluation used a 16,384-token model context, a 2,048-token output ceiling, and at most one retry. Both stages used BF16 inference, FlashAttention 4, eager execution, strict guided JSON decoding, disabled thinking, temperature 0, top- $p = 1$, and seed 42.

Scope	Judge+	A_+	A_-	Overall	κ
All	2.9	69	100	99.1	0.81
FinanceBench	2.0	63	100	99.2	0.77
LegalBench-RAG	3.0	66	100	99.0	0.79
MultiHop-RAG	3.2	71	100	99.1	0.82
QMSum	1.5	55	100	99.3	0.70
TAT-QA	2.1	85	100	99.7	0.91

Table 3: Human audit of Stage 2 support decisions ($n = 200$; 40 per dataset). Judge+ is the production share of positive decisions (%). A_+ and A_- are agreement on judge-positive and judge-negative decisions, reweighted to production frequencies; Overall and κ are prevalence-weighted. Per-dataset rows rest on about 20 audited decisions per label and are indicative only.

B Model and Retrieval Configuration

Neural retrieval. Table 4 summarizes the six neural retrievers. Prefixes are concatenated directly with the input. Dense embeddings are L2-normalized and searched using exhaustive float32 cosine similarity with TF32 disabled. Late-interaction models use L2-normalized token embeddings and MaxSim.

Semantic chunking. Semantic-512 uses jina-embeddings-v5-text-nano-text-matching. Three-sentence windows are encoded with the Document: prefix, an 8,192-token model limit, 768-dimensional normalized embeddings, BF16, and FlashAttention 4. Gradient-based boundaries use a sentence buffer of one. The resulting source-exact chunks have zero overlap and are hard-capped at 512 o200k_base tokens.

BM25 and fusion. BM25 uses BM25S with Lucene term weighting and IDF, $k_1 = 1.2$, and $b = 0.75$. Text is normalized with NFKC, Unicode case-folded, and split into Unicode alphanumeric runs. We apply neither stemming nor stop-word removal. Each hybrid condition combines the saved BM25 and neural top-20 rankings using equal-weight reciprocal-rank fusion:

$$s_{\text{RRF}}(u) = \sum_{m \in \{\text{BM25}, \text{neural}\}} \frac{1}{60 + \text{rank}_m(u)}.$$

Late-interaction retrieval. Both ColBERT models use FastPLAID with 4-bit residuals, four k-means iterations, seed=42, n_ivf_probe=8, n_full_scores=8192, and no token pooling. Dense retrieval performs an exhaustive corpus scan, whereas FastPLAID candidate generation is approximate; we discuss this implementation asymmetry as a limitation.

Table 4: Neural retriever models and representations.

Model	Representation
EmbeddingGemma-300M	Native pooling; 768-dimensional L2-normalized embedding
Qwen3-Embedding-0.6B	Last-token pooling; 1,024-dimensional L2-normalized embedding
GTE-ModernBERT	CLS pooling; 768-dimensional L2-normalized embedding
LFM2.5-Embedding-350M	CLS pooling; 1,024-dimensional L2-normalized embedding
GTE-ModernColBERT	128-dimensional L2-normalized token embeddings with MaxSim
LFM2.5-ColBERT-350M	128-dimensional L2-normalized token embeddings with MaxSim

Reranking We use the GTE, Qwen3, and Nemotron rerankers (Table 2). Each reranker scores only the raw question and a retrieved chunk and preserves the first-stage top-20 candidate set. GTE uses the tokenizer’s question–document pair format with an 8,192-token limit. Nemotron uses question:{query} \n \n passage:{document} with an 8,192-token limit. Qwen uses its retrieval instruction and scores $\text{logit}(\text{yes}) - \text{logit}(\text{no})$ with a 32,768-token limit. Over-length inputs remove document tokens before question tokens.

C Complete Per-Configuration and Per-Model Results

This appendix reports the complete model-level results underlying the aggregated analyses in Section 4. All displayed values use the arithmetic mean of question-level claim recall across the five context capacities (256–4,096 tokens). Figure 6 is descriptive; Figures 7–11 report paired component substitutions.

Reranker	Dataset	Chunker	Retriever	Δ (pp)
GTE	FinanceBench	Fixed-256	EmbeddingGemma	−0.40
GTE	FinanceBench	Fixed-256	Qwen3 Emb.	−2.00
GTE	FinanceBench	Sentence-512	Qwen3 Emb.	−1.20
Qwen3	TAT-QA	Fixed-256	GTE ColBERT	−0.02

Table 5: The four negative reranker effects among the 780 cells. Δ is the paired change in claim recall relative to no reranker, averaged across the five capacities.

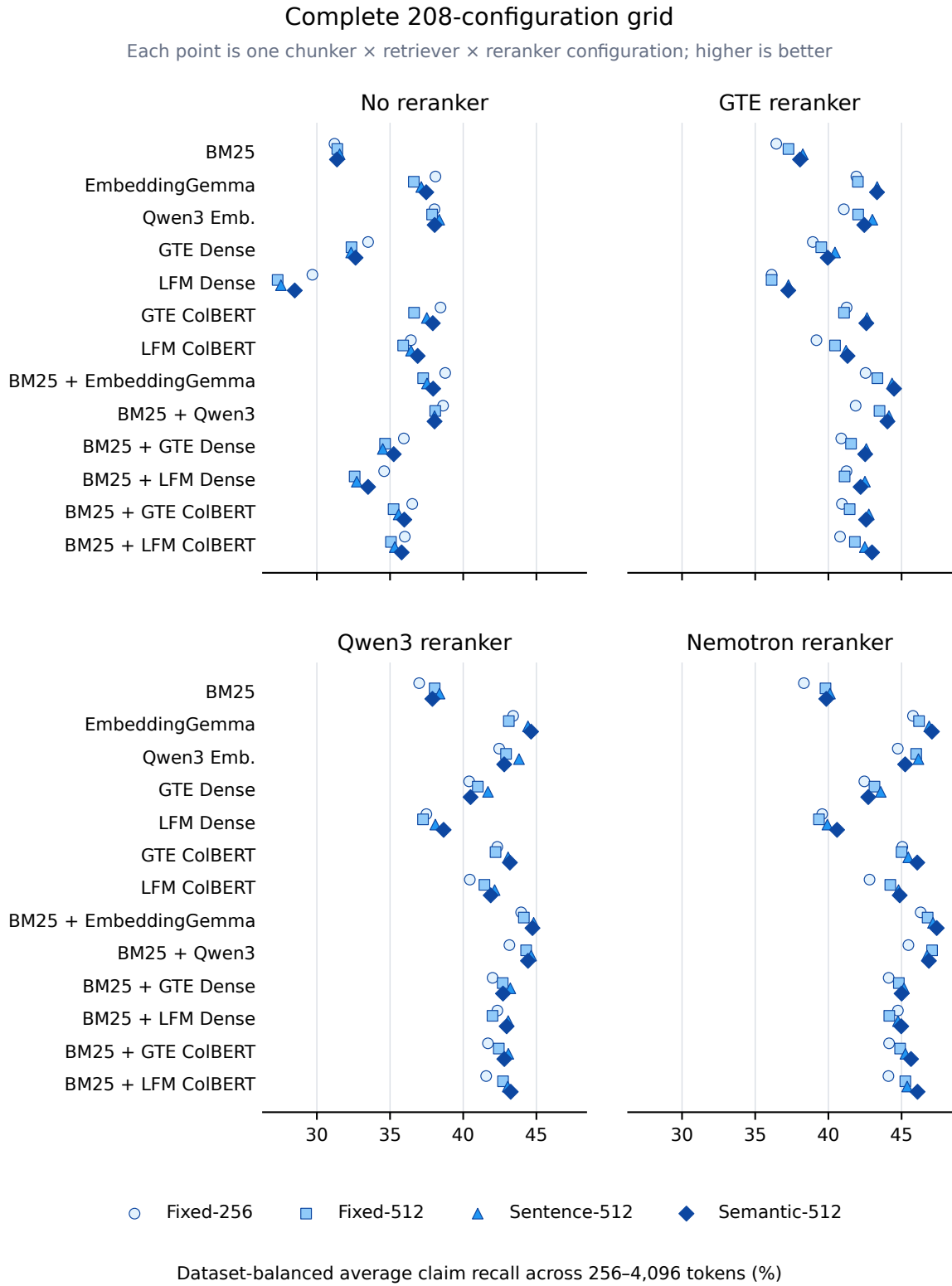


Figure 6: **Descriptive results for all 208 pipeline configurations.** Each point represents one chunker–retriever combination within the indicated reranking condition. Values are dataset-balanced average claim recall across the five context capacities. This figure describes complete-pipeline performance and is not interpreted as a paired component effect.

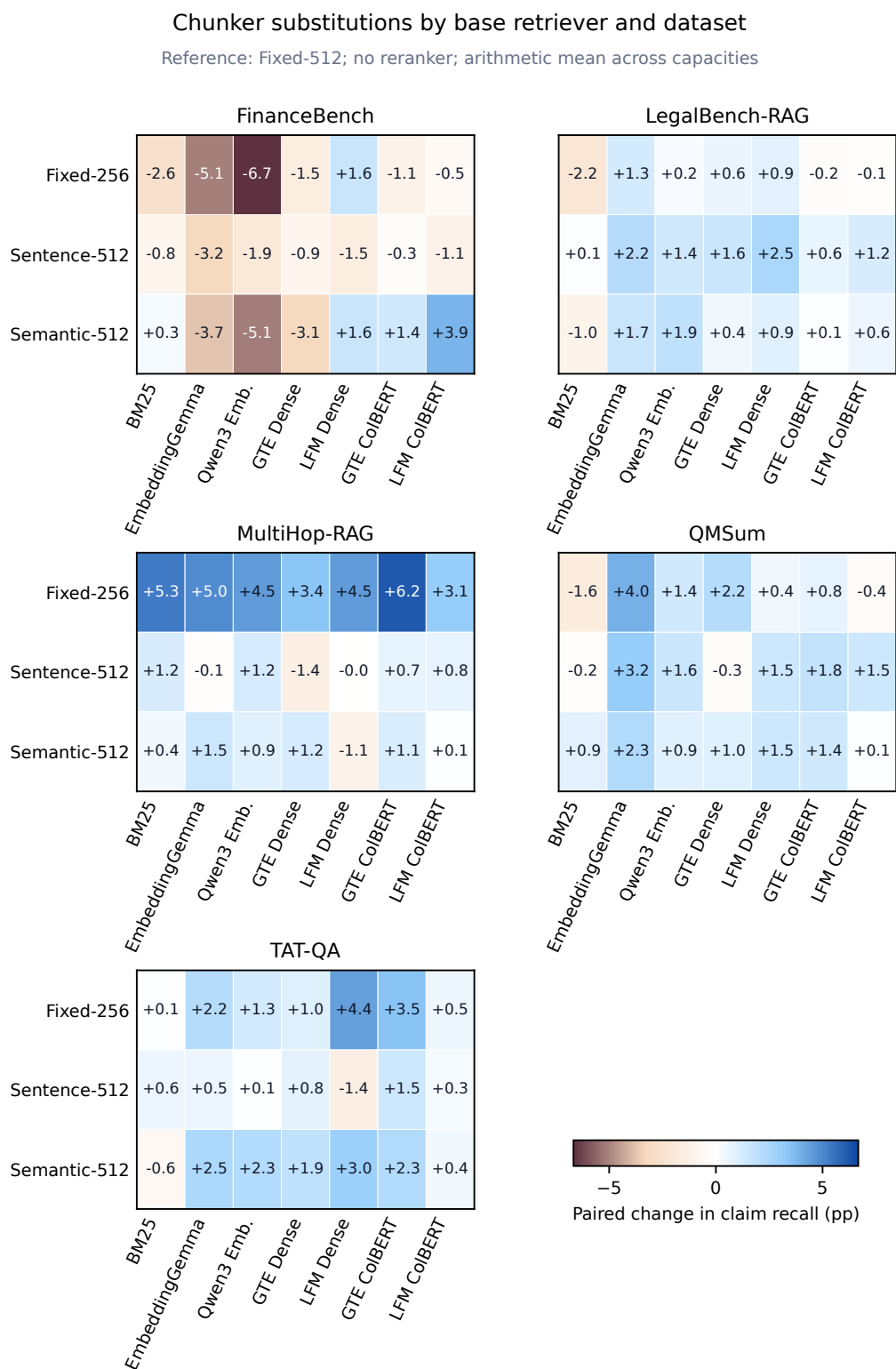


Figure 7: **Chunker effects for every dataset and base retriever.** Each cell reports the paired change in claim recall from replacing Fixed-512 with the indicated chunker, while holding the dataset, retriever, capacity, and no-reranker condition fixed. Values are percentage-point effects averaged equally across capacities.

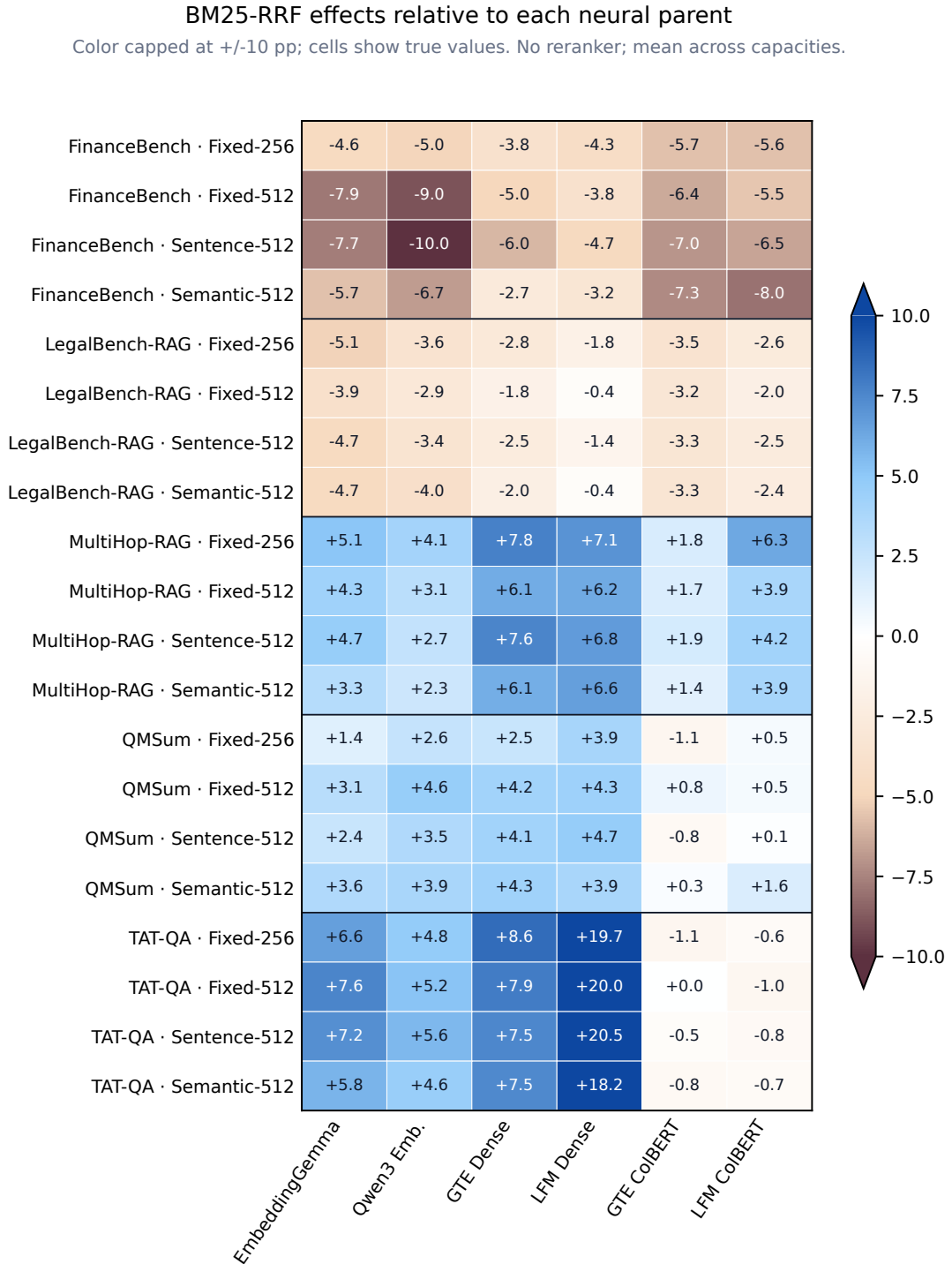


Figure 8: **All 120 matched BM25-RRF fusion effects.** Each cell compares a BM25-RRF retriever with its exact neural parent under the same dataset, chunker, capacity, and no-reranker condition. Values are paired percentage-point changes in claim recall, averaged equally across capacities. Colors are capped at ± 10 points so that smaller effects remain visible; cells show the true values.

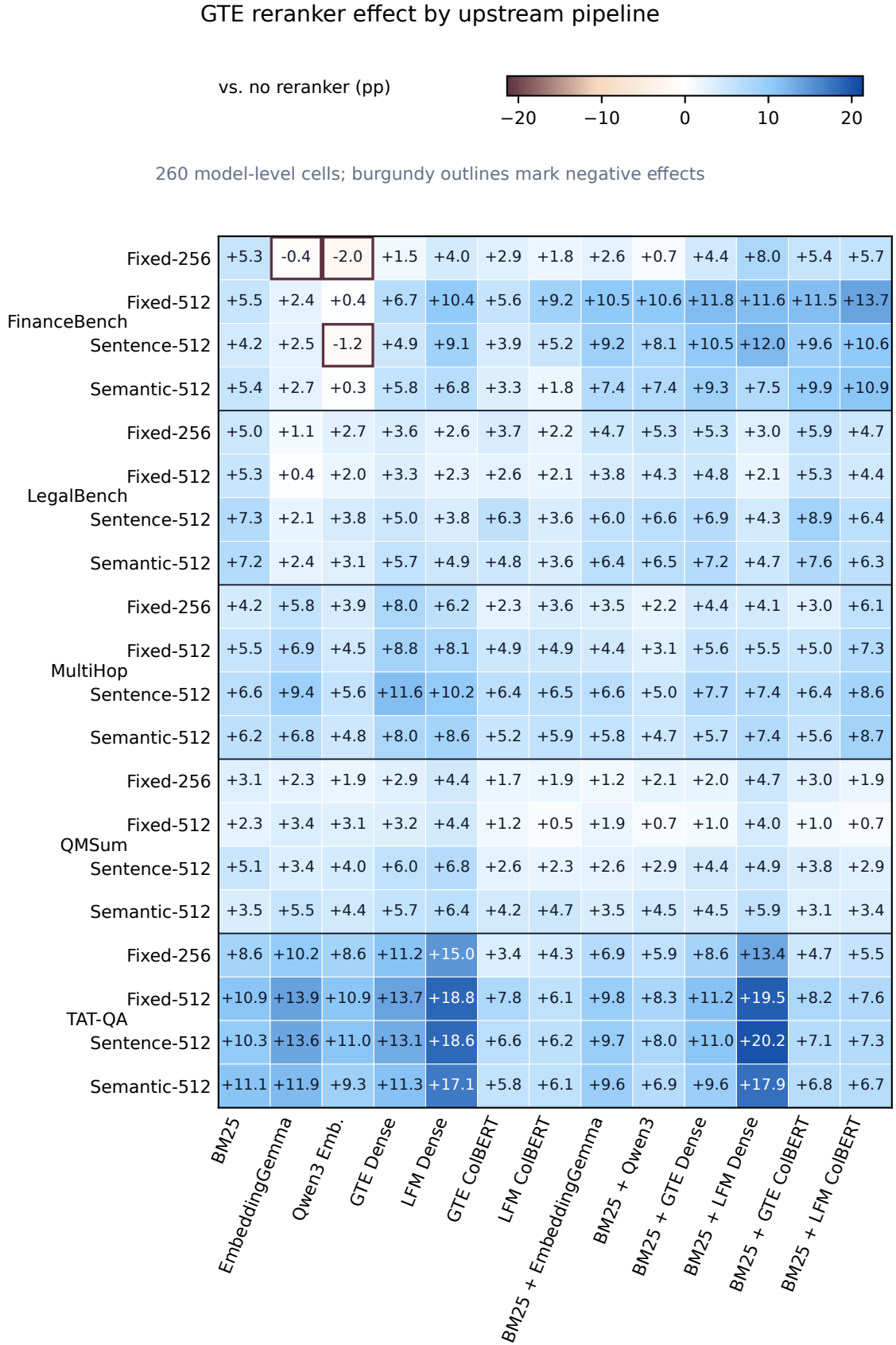


Figure 9: **GTE reranker effects across 260 upstream cells.** Each cell is the paired percentage-point change in claim recall relative to no reranker for the same dataset, chunker, and retriever, averaged equally across the five token capacities. Burgundy outlines mark negative effects.

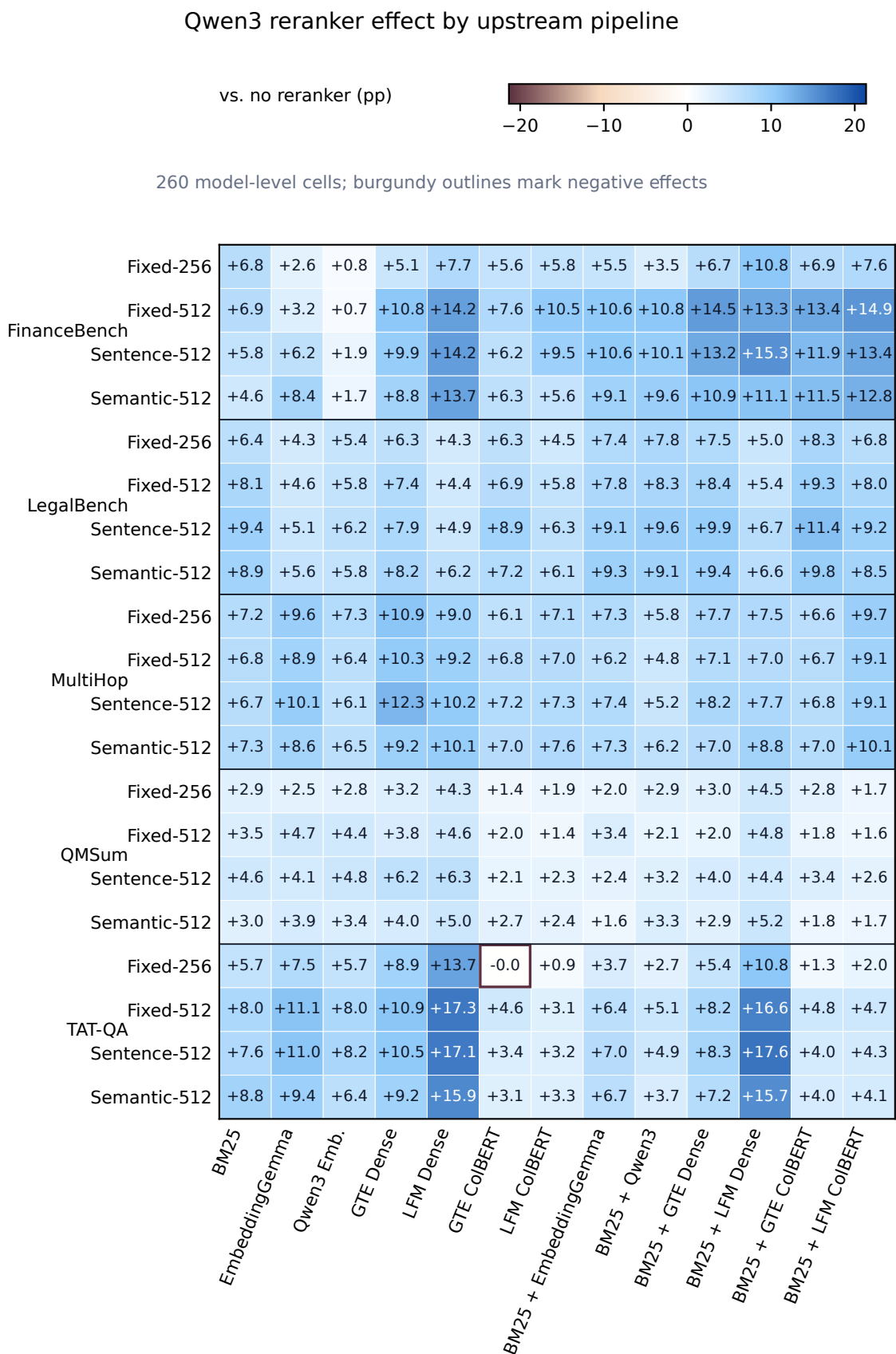


Figure 10: **Qwen3 reranker effects across 260 upstream cells.** Cells use the same paired comparison and capacity averaging as Figure 9. Burgundy outlines mark negative effects.

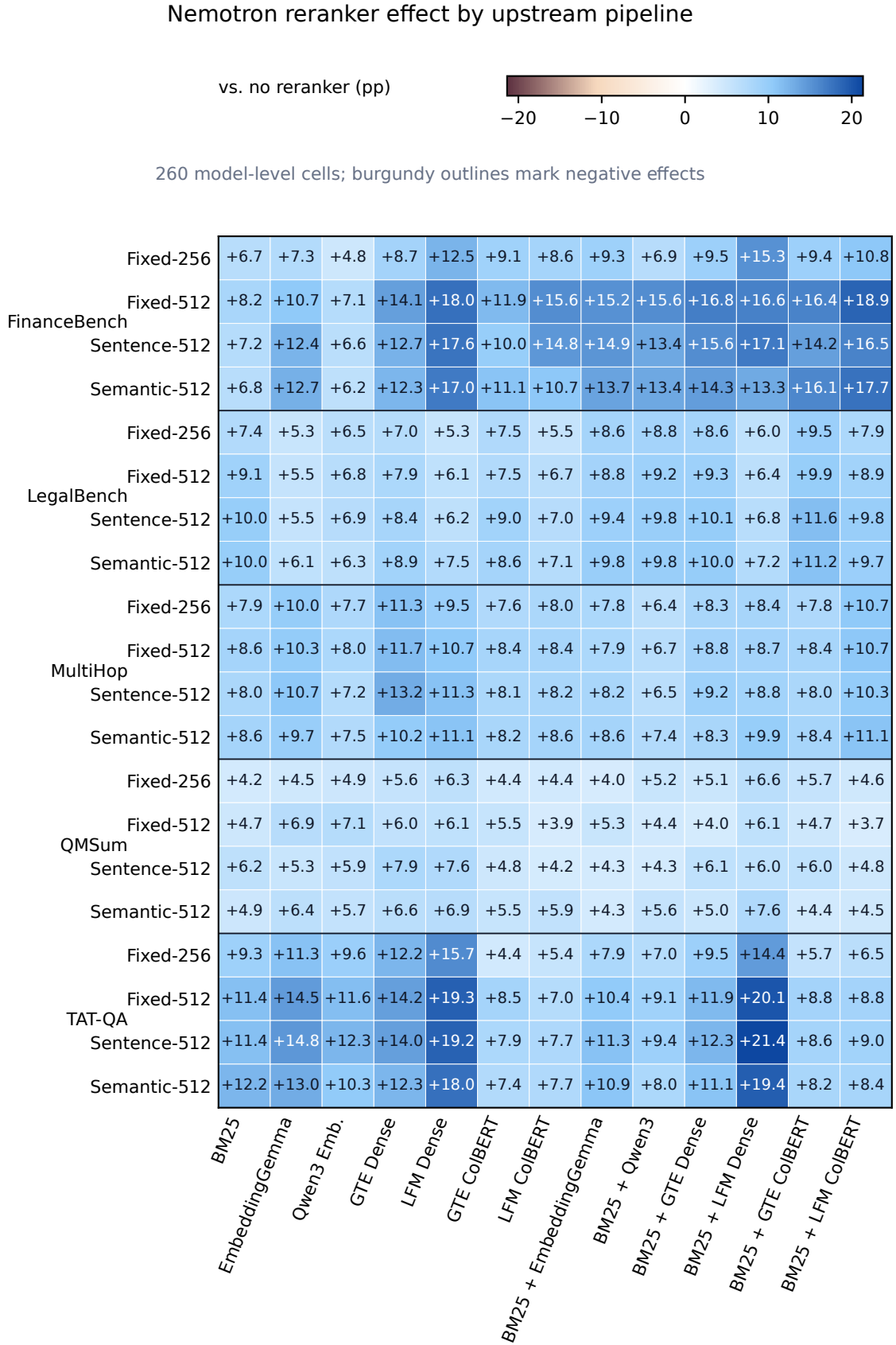


Figure 11: **Nemotron reranker effects across 260 upstream cells.** Cells use the same paired comparison and capacity averaging as Figure 9. Across these three figures, 776 of the 780 effects are positive; burgundy outlines identify the four negative cells.