CompEvidenceBench: Diagnosing Complementary Evidence in Context Compression

Anonymous authors
Paper under double-blind review

Abstract

Budgeted context compression is usually evaluated only by end-to-end answer accuracy. This aggregate score cannot reveal whether failure reflects unavailable evidence, a compressor that drops a jointly useful context, or a reader that does not use retained evidence. We introduce **CompEvidenceBench**, an auditable benchmark for reader-conditioned complementary-evidence selection failures. For each question, it fixes a budgeted candidate slate containing closed edit quartets, and uses an offline interaction score to audit whether a joint edit is more useful than its constituent edits under a fixed reader. Reference answers, candidate utilities, oracle choices, and interaction labels are never visible to the compressor. On HotpotQA-derived candidate slates at 512 and 1024 tokens, an answer-dependent candidate oracle is 13.18 and 13.04 F1 points above a frozen additive selector overall; on reader-conditioned complementary cases, the gaps are 41.13 and 33.62 points. Yet a three-seed direct-text selector with interaction supervision does not recover this headroom reliably, and a separate frozen pairwise-decision track also yields no stable gain. In contrast, an external generative compressor improves on the complementary subset under both the construction reader and a Mistral cross-reader, although its unknown training overlap prevents a leakage-free ranking claim. CompEvidenceBench therefore separates *available opportunity*, *learned recovery*, and *external sensitivity*: it is an auditable diagnostic testbed rather than evidence for a solved compression method. The anonymized release provides the candidate slates, training and development inputs, offline audit labels, frozen protocol manifests, and a script that regenerates every table; it is available at https://anonymous.4open.science/r/CompEvidenceBench-2434/README.md.

1 Introduction

Retrieval-augmented generation and long-context language models must often compress an evidence pool before answering a question (Sharma et al., 2025; Liu et al., 2024). Existing evaluation typically asks whether the final retained context yields a correct answer under a token budget. This aggregate outcome is necessary, but it is insufficient for diagnosing a compositional failure: evidence a and b may each appear expendable in isolation, while their joint presence is needed to establish the answer. A compressor that values evidence only by isolated utility has no direct signal for retaining such a pair.

CompEvidenceBench fixes, for each example, a question, an evidence pool, a finite slate of budget-feasible contexts, and a reader. The compressor sees only the question, evidence, slate, and budget; it never sees a reference answer, reader utility, or complementarity label. An offline audit constructs **closed edit quartets**—an anchor, two singleton edits, and their joint edit—so that interaction can be measured while the deployed task remains an ordinary answer-F1 evaluation. Figure 1 gives the intuition: a plausible singleton can survive compression while a jointly useful pair is lost, and only the fixed downstream evaluation distinguishes the two outcomes.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

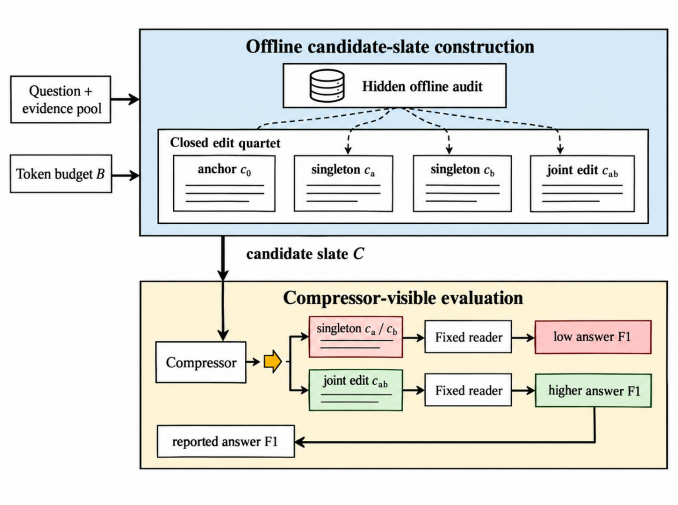


Figure 1: CompEvidenceBench as a controlled selection task. The compressor receives only q, E, B, and the budget-feasible candidate slate $\mathcal{C}$. Offline audit values construct and diagnose closed edit quartets but are hidden from the compressor. Retaining a plausible singleton can yield low answer F1, whereas retaining a jointly useful edit can yield higher F1 under the same fixed reader.

We evaluate three diagnostic questions. *Can a fixed candidate slate expose budget-feasible, reader-conditioned complementary opportunity? Can the protocol distinguish available opportunity from learned selection behavior under matched controls? Do fixed retained contexts remain distinguishable under a second reader?* We answer the first using an answer-dependent diagnostic oracle, the second with frozen additive and shuffled-label controls, and the third by replaying fixed contexts through an independently configured Mistral reader. The oracle measures what is available in the slate; the controls and reader replay test what the evaluated compressors retain.

On opened HotpotQA-derived development slates (Yang et al., 2018), the candidate oracle has substantial headroom over a frozen additive selector, especially on reader-conditioned complementary cases. In contrast, a direct-text learner with explicit interaction supervision is not reliably better than a within-question shuffled-label control across three seeds. An external generative compressor improves on the same diagnostic subset under the primary reader and Mistral, showing that the protocol can distinguish methods; its unknown pretraining overlap, however, rules out a leakage-free SOTA claim.

The contribution is an **executable benchmark and diagnosis**, not a new state-of-the-art compression algorithm. Following prior benchmark work, we specify the task contract and report the controls alongside the baseline results (Cao et al., 2025; Fatahi Bayat et al., 2025; Kwan et al., 2024):

- We define a **controlled, reader-conditioned diagnostic** with closed edit quartets, strict token budgets, answer-hidden compressor inputs, and matched non-complementary controls.
- We provide an executable protocol that separates construction-time audit labels, candidate-set opportunity, learned selection, and reader evaluation.
- We characterize opportunity, learned recovery, and external sensitivity separately: three-seed and shuffled-label controls test the learned selector, while a cross-reader external baseline tests discriminability without being treated as a leakage-free ranking result.

2 Related Work

Retrieval and compression. Dense retrieval work spans hierarchical, multi-task, weak- and distant-supervision, table-aware, and attribution-oriented choices (Karpukhin et al.,

2020; Ni et al., 2022; Khattab & Zaharia, 2020; Santhanam et al., 2022; Liu et al., 2021; Li et al., 2021; Zhao et al., 2021; Nian et al., 2025; Herzig et al., 2021; Cheng et al., 2023; Li et al., 2022), together with query generation, negative sampling, and rank stratification (Rajapakse, 2023; Zhong et al., 2022; Zhuang et al., 2023; Rajapakse et al., 2024; Ma et al., 2024; Zhuang et al., 2022; Reichman & Heck, 2024; Le Vu et al., 2023). For the compression stage, prompt, key–value cache, long-context, and attention-efficiency methods trade length against downstream utility (Izacard & Grave, 2021; Jiang et al., 2023; 2024; Cheng et al., 2024; Chevalier et al., 2023; Bertsch et al., 2023; Packer et al., 2023; Gutierrez et al., 2024; Dao et al., 2022; Corallo & Papotti, 2024; Li et al., 2025). These motivate a diagnostic that separates individually useful evidence from useful combinations.

Compression as selection. The closest line treats compression as explicit selection over retrieved material: context pruning scores spans or sentences for retention under a target length (Chirkova et al., 2025), selective augmentation decides what to keep and what to abstract (Xu et al., 2024), and self-information filtering removes low-value sentences without a trained selector (Li et al., 2023). These are regression or filtering objectives over *individual* units, evaluated by end-to-end answer quality. CompEvidenceBench keeps that selection contract but changes the measurement target: it asks whether a retained unit is useful only in combination with another, and audits that interaction through closed counterfactual quartets under a fixed reader, using a length-matched PROVENCE-native configuration as the recall-side reference point (Section 5.1).

Set-level selection objectives. A separate line scores a *set* rather than individual units: maximal marginal relevance trades relevance against redundancy (Carbonell & Goldstein, 1998), submodular coverage formalizes representative-plus-diverse selection with greedy guarantees (Lin & Bilmes, 2011; Nemhauser et al., 1978), and determinantal point processes model repulsion among selected items (Kulesza & Taskar, 2012). These are our nearest methodological relatives, since in each a unit's value already depends on what else is selected. They differ in what is fixed in advance—each optimizes a *prior* set function, whereas CompEvidenceBench defines complementarity *outcome-first* as a closed-quartet contrast under a fixed reader—and a diversity term can reduce redundancy without making any retained pair answer-producing, the failure the audit exposes.

Long-context and multi-hop evaluation. Existing long-context suites test broad task mixtures and stress conditions over supplied contexts (Bai et al., 2024; Zhang et al., 2024; Hsieh et al., 2024; Dong et al., 2024; Pfister & Hotho, 2024; Cao et al., 2025; Fatahi Bayat et al., 2025; Kwan et al., 2024; Gu et al., 2024; Huang et al., 2025; Churin et al., 2025; Qiu et al., 2024), while multi-hop QA supplies compositional evidence requirements (Yang et al., 2018; Ho et al., 2020; Trivedi et al., 2022; Khot et al., 2020; Ferguson et al., 2020; Kwiatkowski et al., 2019; Joshi et al., 2017; Rajpurkar et al., 2016; Feldman & El-Yaniv, 2019; Aghaebrahimian & Jurčíček, 2016; Das et al., 2019; Chen et al., 2020; Xiong et al., 2019; Beauchemin et al., 2024; Li, 2021; Gong & Bowman, 2018). They differ in the capability tested and the information given a model, but CompEvidenceBench instead fixes the candidate slate and reader, permitting an auditable counterfactual about whether a feasible jointly useful context was retained.

Closest distinction. Broad retrieval and attribution benchmarks measure end-to-end retrieval, embedding, knowledge, or citation behavior (Muennighoff et al., 2023; Petroni et al., 2021; Gao et al., 2023; Yan et al., 2024; Jeong et al., 2024); reasoning methods instead improve intermediate search or answer construction (Wei et al., 2022; Yao et al., 2023). Neither family fixes a finite, budget-feasible candidate slate and then asks whether two individually weak edits become useful only together. Our evaluation concerns the preceding context-selection step and asks for six components jointly: token-budgeted selection, a fixed finite candidate slate, closed counterfactual edits, an outcome-based interaction audit, **answer-derived audit values** hidden from selection, and fixed-context reader replay. Broad long-context suites such as LongBench (Bai et al., 2024) and RULER (Hsieh et al., 2024) supply none of these; prompt-compression methods such as LLMLingua (Jiang et al., 2023) and LongLLMLingua (Jiang et al., 2024) control retained length but do not supply

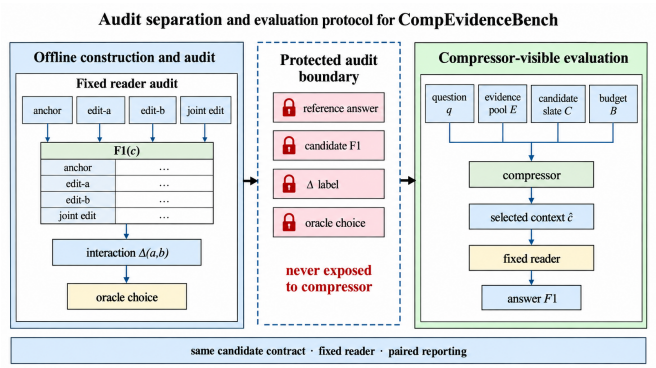


Figure 2: Audit separation and evaluation protocol. Fixed-reader outcomes over each closed quartet produce candidate F1, interaction, and oracle values during offline construction. These answer-derived fields remain behind the protected audit boundary; the compressor receives only the question, evidence pool, candidate slate, and budget before the fixed reader reports answer F1.

the counterfactual audit contract needed to separate candidate availability from learned selection.

3 Benchmark Design

3.1 Task contract and access rules

For question q, evidence pool E, and token budget B, a compressor chooses one context c from a finite candidate slate $\mathcal{C}(q, E, B)$. A fixed reader produces an answer $r(q, c)$, scored against a reference answer using normalized HotpotQA F1. The compressor may inspect q, E, candidates, and B, but may not access reference answers, candidate reader outputs, candidate F1 values, or interaction labels.

The **finite slate** makes counterfactual analysis possible without confusing generation quality with selection quality, and it makes the diagnostic oracle explicit: for each question the oracle selects the maximum-F1 candidate, which is answer-dependent and measures only the headroom available in the fixed slate. As Figure 2 depicts, this separation is enforced operationally: offline audit values flow into construction and analysis, while the compressor receives only the evidence, candidate slate, and budget.

3.2 Closed edit quartets and complementarity

Starting from an anchor context, we construct two feasible atomic edits a and b and their joint edit ab, all satisfying the same budget. Let $F(c)$ be the fixed reader's answer F1 for context c. We define the offline interaction score

$$\Delta(a, b) = F(ab) - F(a) - F(b) + F(\text{anchor}). \tag{1}$$

We call an example *reader-conditioned complementary* when its slate contains a closed quartet with positive Δ under the fixed construction reader. The label is an **audit-only construction attribute**: it is not released to a compressor at inference time. We also create matched non-complementary controls with no positive-Δ quartet, matching candidate-count and budget strata. The stratified evaluations report the complete population (All), complementary cases, and matched controls.

This operational label is not a reader-independent semantic property, nor a claim that Δ is the only definition of complementary evidence in natural language. It creates a controlled and auditable setting in which the failure to retain jointly useful content can be measured under a stated reader and candidate contract.

A concrete case makes the contrast visible. For the released question *"Who directed the film in which Joseph Pilato played Captain Rhodes?"*, the pool contains both *"He is perhaps*

Table 1: Population accounting for the released prototype. Counts are question–budget instances; development strata are subsets of the 1,240 opened-development instances.

Population	**512**	**1024**
Training instances	2,480	2,480
Opened development (All)	620	620
Complementary	90	137
Matched control	90	137

best known for his performance as Captain Rhodes in the 1985 film 'Day of the Dead'" and *"Day of the Dead is a 1985 American horror film written and directed by George A. Romero"* in two different documents. Neither answers the question alone: the first supplies the role-to-film link, the second the film-to-director link, and the reference answer *George A. Romero* appears only in the second. Under isolated-utility scoring both are plausible additions, so a budget-limited anchor may keep the first—the more on-topic one, since it names the queried actor—and drop the second, and no reader capability then recovers the answer. This is the structure the quartet contrasts detect, and what the oracle headroom in Section 5.1 aggregates over 620 questions per budget.

3.3 Candidate slate construction

The slate fixes what the diagnostic can expose, so we specify it here; Appendix B gives the full construction. The evidence pool is the HotpotQA question together with its supplied documents expanded sentence by sentence over all ten documents, with no retrieval or relevance filtering, so the pool is a superset of anything a selector may retain. The **anchor** is not a retrieved top-k list: it is the terminal feasible state of a single rollout of the frozen additive selector that also serves as FSR, so quartets are measured around a context a deployable additive method actually produces.

A candidate is a retained-sentence set obtained from the anchor by atomic edits that add one pool sentence. The released prototype enumerates the additive direction—the anchor, every single-sentence addition, and every unordered pair of additions—under a pool bound of four source sentences and a slate bound of eleven candidate sets. A candidate enters the slate only if its *exact rendered length* falls within budget: candidates are rendered by grouping retained sentences by document with a document header, and feasibility is counted on that string under the construction tokenizer rather than estimated. For every candidate differing from the anchor by two edits, the constructor forms both singleton edits and requires their retained-index tuples to be present in the same slate; a quartet missing either singleton is not assigned an invented score.

Because the slate is finite and enumerated around an additive anchor, the opportunity measured against the oracle is a *lower* bound on what a larger edit space could expose: at least the enumerated opportunity exists and was not recovered, which does not upper-bound all compressed texts.

3.4 Data splits, readers, and audit protocol

The current prototype is derived from HotpotQA training questions. The training split contains 4,960 question–budget instances. The opened development split contains 1,240 instances, balanced across 512 and 1024 token budgets (620 each). It includes 227 complementary cases and 227 matched controls: 90 per class at 512 tokens and 137 per class at 1024 tokens. Splits are **disjoint by original question ID**, so budget variants of a question never cross splits.

Table 1 records the denominators governing the main comparisons. A budget is an evaluation condition, whereas the original question ID is the unit for split assignment and paired uncertainty estimation; the complementary and matched-control rows sit beside the full population because a slice result is informative about the diagnostic but cannot replace the All result, and the unequal complementary counts across budgets must remain visible.

Table 2: CompEvidenceBench executable contract.

Component	Contents and access rule
Compressor input	Question, evidence pool, finite candidate slate, budget.
Hidden audit labels	Reference answer, candidate reader F1, Δ, oracle choice. Never exposed to the compressor.
Evaluation output	Fixed-reader normalized answer F1 and tokenizer-specific context length.
Required strata	All examples, positive-Δ complementary examples, matched non-complementary controls.
Statistical unit	Question ID; average optimization seeds within question before paired resampling.

The primary reader, **Qwen2.5-7B-Instruct**, is held fixed throughout construction and development evaluation; its decoded configuration and artifact hashes are recorded in the Reproducibility Statement. To test whether conclusions depend entirely on that reader, we also replay fixed selected contexts through a **Mistral-7B-Instruct-v0.3** cross-reader. The cap uses the **construction tokenizer**; Mistral replay may exceed its native token budget (Appendix F).

3.5 Benchmark contract and audit separation

The public-facing compressor interface contains no outcome-derived attribute, as specified component by component in Table 2. Audit information is separated because exposing it would turn the benchmark into an answer-conditioned ranking exercise rather than a context-compression task. The evaluator may assess a completed action, but the compressor must not receive the evaluator state as an input.

Statistics and controls. We report answer F1 in points, resample question IDs with replacement 20,000 times for paired contrasts, and report percentile 95% intervals, averaging seeded selector outputs within question first. Percentile intervals at this stratum size are discrete rather than smooth, so overlapping intervals are read as *no detected difference*, not as evidence of equivalence, and they describe uncertainty about this population rather than about arbitrary new questions. Development results are **adaptive** and are not called an independent test. Three protocol controls bound the negative results: a **matched-control** stratum supplies a non-positive counterpart for every positive-Δ example at the same budget and matched candidate count, so a slice change is not read as a general gain; a **label-causality** control permutes Δ targets within question at fixed target scale, label count, architecture, optimizer, and training budget—this changed labels for 1,413 of 1,517 questions, yet for seed 42 the correct and shuffled variants selected the same candidate position on all 1,240 development questions; and a **reader-robustness** check holds compressed contexts fixed across readers, so a change in answer quality cannot be a change in what was selected.

Scope of the test. CompEvidenceBench evaluates *selection under a finite candidate slate*, not evidence synthesis or the retrieval of missing documents; generative compressors are permitted, but their output is judged under the same reader and budget contract. Because the interaction score is computed only after reference answers exist, it is never a test-time target. Appendix J states in full what the benchmark does and does not license.

4 Compressors Under Test

4.1 Frozen additive reference

FSR-additive. Our frozen reference selector scores candidate evidence by an additive residual policy. It is the principal non-generative baseline because it makes the independent-scoring assumption that CompEvidenceBench probes.

Table 3: Seed-42 primary-reader opportunity and learned recovery on opened development data ($n = 620$ per budget; 90 and 137 complementary/matched-control questions at 512/1024 tokens). "Set-aware" is the correct-label v2 selector; Oracle selects the highest-F1 candidate from the reference answer and is diagnostic only. The direct-text ablation is in Appendix Table 8.

	512 tokens			1024 tokens		
Stratum	**FSR**	**Set-aware**	**Oracle**	**FSR**	**Set-aware**	**Oracle**
All ($n = 620$)	65.35	64.31	78.54	64.75	63.61	77.79
Complementary ($n = 90/137$)	53.69	64.63	**94.82**	55.07	60.52	**88.69**
Matched control ($n = 90/137$)	68.73	65.22	–	69.95	66.38	–

4.2 Learned selectors

Text selectors. Direct-text v3 scores BGE-M3 candidate encodings with a scalar head; its ablation uses seeds 42, 43, and 44. Set-aware v2 instead adds two candidate-set self-attention layers and a residual over frozen FSR scores. Both recipes use utility regression and choice loss; interaction arms also regress closed-quartet Δ, and within-question shuffled targets control that supervision. Appendix Table 8 and Appendix D report direct-text v3 per seed; Appendix C specifies the objectives and settings.

4.3 External generative compressor

Generative compressor. We evaluate an already available BRIEF-Pro automatic hard-cap compressor as a strong external baseline, on exactly the same question IDs, budgets, readers, and strata. Its public documentation reports HotpotQA training among synthetic-training seeds, but exact question-level overlap with our train-derived population is unknown, so we report it as a strong external baseline and never as a leakage-free SOTA comparison.

5 Findings

5.1 Candidate slates expose reader-conditioned complementary opportunity

A difficult subset is only informative if some feasible compressor can succeed on it, so we compare FSR with an answer-dependent best-candidate oracle on exactly the same slate. On all examples the oracle is 13.18 and 13.04 F1 points above FSR at 512 and 1024 tokens; on reader-conditioned complementary cases the gap is 41.13 and 33.62. Under the fixed slate and reader, the diagnostic subset therefore contains answer-producing contexts that meet the stated budget. This is an **opportunity result**: it does not mean a deployable compressor can access the oracle context.

The complementary-only set-aware improvement must not be read in isolation: this slice is defined by the same primary reader, the full-population result is negative, and the matched controls degrade. That divergence is the diagnostic role of the benchmark—aggregate F1 alone would hide the large behavior change on complementary cases, while the control degradation blocks reading the slice gain as a general improvement.

5.2 Where the budget actually goes

A diagnostic subset is worth reporting only if failures trace to what the compressor retained rather than to a reader unable to use evidence, and only if the retained content is competitive with a length-matched baseline. We test both.

The failures are repairable by recall. We run a length-matched intervention: a candidate context is replaced by one containing the annotated gold-support sentences, with rendered length held fixed under the construction tokenizer, and the primary reader is re-run. Restoring annotated support raises answer F1 by +26.03 points over a length-matched Provence-native baseline at 512 tokens ($n = 73$, 95% CI $[15.07, 36.99]$) and +26.39 at

Table 4: BRIEF-Pro minus FSR answer-F1 points on opened development data ($n = 620$ per budget; 90/137 complementary and matched-control questions at 512/1024 tokens). Brackets are paired 95% bootstrap intervals; † marks intervals including zero.

Reader	Stratum	512 tokens	1024 tokens
Primary	All	+12.64 [9.43, 15.92]	+8.88 [5.74, 12.10]
Primary	Complementary	**+17.65 [7.66, 27.74]**	**+14.77 [6.94, 22.67]**
Primary	Matched control	+10.48 [2.02, 19.17]	+4.49 [-0.58, 9.78] †
Mistral	All	+7.80 [5.01, 10.61]	+9.30 [6.40, 12.21]
Mistral	Complementary	**+10.57 [3.83, 17.52]**	**+14.96 [8.82, 21.37]**
Mistral	Matched control	+9.49 [1.74, 17.48]	+5.89 [-0.22, 12.04] †

1024 ($n = 72$, $[15.97, 36.81]$; 10,000 paired resamples), while length-matched relevance-only and random controls stay near 50 and 15 F1. The binding constraint is therefore what is retained, not a reader unable to use support once present. This establishes repairability only: the intervention is injected rather than selected, so it does not show a learned pairwise model is superior.

The retained content is not the bottleneck. The natural alternative is that the slates are simply weaker than a modern length-matched pruning baseline, so we score our released slate selections against a length-matched PROVENCE-native configuration on the same reused development questions, original reader, and final budgets, with question-level paired bootstrap and unadjusted descriptive intervals. The additive selector is +3.82 F1 above it at 512 tokens ($[-0.25, 7.94]$) and +0.64 at 1024 ($[-3.04, 4.27]$); its pairwise and conditional variants are +5.91 ($[1.53, 10.32]$) and +4.43 ($[0.63, 8.27]$) at 512, and +1.27 ($[-2.31, 4.90]$) and +0.16 ($[-3.69, 3.91]$) at 1024. Purely embedding-derived initial-rank selectors, which use no learned utility head, score below the same baseline in all four cells (-5.42, -7.78 at 512; -2.59, -11.44 at 1024; three intervals exclude zero). The slates are thus competitive with, and not consistently worse than, a length-matched neural pruner here, so the negative results below are not an artifact of an uncompetitive slate. These comparisons reuse opened development questions and are descriptive, not an independent confirmation.

5.3 Interaction supervision does not trivially recover the diagnostic opportunity

Across three development seeds, direct-text v3 averages below FSR at both budgets: FSR reaches 65.47 and 65.00 F1 at 512 and 1024 tokens, while utility-only, shuffled-label, and correct-label arms all land between 63.76 and 64.28 (Appendix Table 8). Correct minus shuffled supervision is +0.31 F1 at 512 tokens and +0.50 at 1024; both paired intervals include zero (Appendix D). This is **recipe-specific negative evidence**, not proof that complementary evidence is impossible to learn.

An earlier interaction track supplies a separate, frozen negative check. It trained pairwise decision models at a 6,200-question scale and evaluated them once on 1,200 previously sealed questions after all model hashes were bound. At 512 tokens, decision-pairwise differed from atomic-additive by -0.11 F1 (95% CI $[-1.39, 1.17]$); at 1024 tokens the difference was -0.55 ($[-1.90, 0.79]$). This track has a different candidate-edit space and is not a confirmation of Set-aware v2, but it reinforces that simply adding pairwise supervision and more training data did not establish stable improvement.

5.4 An external generative compressor serves as a sensitivity probe

BRIEF-Pro's paired improvements over three-seed FSR remain positive on reader-conditioned complementary examples under both readers and both budgets (Table 4). This shows that the benchmark can distinguish methods and that the diagnostic subset is not intrinsically unanswerable. Because training overlap is unknown and all these examples are opened development data, these results are evidence of **external sensitivity**, not a leakage-free generalization or ranking claim.

6 Discussion

6.1 Why the interaction signal is weak here

The negative result is not specific to one architecture. Three properties of the released data each bound how much an explicit correction term can contribute, and each is checkable in the release. First, at 512 tokens roughly **87.8% of question-weighted Δ targets are exactly zero**, so interaction supervision is sparse and heavily imbalanced, and at target level predicting zero is competitive with every trained arm (Table 10). Second, the candidate encodings are question-conditioned and already carry global context, so part of the signal an interaction term is meant to add may be absorbed into the representation, leaving the head to fit a residual rather than a first-order effect. Third, selection is an argmax over a slate with a narrow utility spread and most position changes are reader-F1 ties, so a small score correction usually changes the selection without changing the answer—visible in the shuffled control, which reproduces all 1,240 selections despite failing at Δ regression. Appendix K develops each hypothesis and states what would falsify it.

6.2 Methodological consequences

Answer F1 alone does not identify the source of a method difference, so the checks above impose three requirements on any future claim: opportunity measured on the same candidate slate and budget as the deployable baseline; learned recovery validated against a frozen additive reference and a matched training control, so a slice gain is not confused with a general improvement; and reader replay that holds the compressed context fixed, so a reader-robustness result is not a change in what was selected. A compressor may score candidate sets jointly, search candidate edits, or generate a compact context, provided selection uses only the public question, evidence, slate, and budget fields; a mechanism claim additionally requires a pre-frozen configuration and question-disjoint confirmation against the additive reference.

A practical consequence is that a single aggregate F1 under-determines what a compressor does, since the same score can arise from retaining more useful pairs, from a stronger reader, or from a different slate; the recipe here transfers to any setting where a base context and independent atomic additions can be enumerated.

6.3 Limitations

CompEvidenceBench is designed to make a missing evaluation axis visible, not to establish that every multi-hop compression error is caused by pairwise complementarity. Within the released HotpotQA-derived slates and the fixed construction reader, the evidence establishes budget-feasible candidate opportunity, tested-recipe failure to recover it reliably, and reader replay for fixed contexts; the labels are reader-conditioned diagnostics, so replaying fixed contexts through a second reader checks method sensitivity rather than turning Δ into a reader-independent semantic label. The single-source prototype motivates rather than establishes cross-dataset generalization, the set-aware model has no same-candidate-space untouched confirmation because it **failed its development gate**, and the frozen pairwise track is an adjacent negative result rather than a confirmation of another method; BRIEF-Pro's unknown training overlap leaves it a sensitivity baseline (Table 6).

7 Conclusion

CompEvidenceBench makes one compression question measurable: when a budget-feasible context is useful only through the combination of its evidence under a fixed reader, does a compressor retain it? The present experiments establish candidate-slate opportunity and show that the tested interaction-learning recipes do not yet recover it reliably, while the same population distinguishes an external generative compressor under two readers. The benchmark is therefore a way to evaluate future compression methods under sealed confirmation and auditable provenance, not evidence that a learned selector already solves this problem.

Reproducibility Statement

Released artifact. The anonymized release contains the HotpotQA-derived candidate slates, training and development inputs, offline audit labels, frozen protocol manifests, the research source archive with host-specific paths and launchers removed, and a script that regenerates every table from the released aggregate values (https://anonymous.4open.science/r/CompEvidenceBench-2434/README.md). No model checkpoint, credential, or server address is included.

Reader provenance. The primary reader is **Qwen2.5-7B-Instruct** in BF16 with SDPA attention and greedy decoding (64 new tokens, batch size 1). Its provenance was bound by hashing the model index, all four referenced weight shards, the model and generation configs, and the tokenizer assets—eleven files—plus a chat-template rendering probe; the inventory hash is in the release manifest. Because these hashes were produced retrospectively, they establish byte identity of the evaluated reader, not pre-registration of the labels built with it. The cross-reader is **Mistral-7B-Instruct-v0.3**, under a separately frozen protocol.

Freezing and outcome-blind auditing. Each formal run binds its decision, audit, and scoring rules by SHA-256 before predictions are scored, and is scored only after an outcome-blind post-run audit passes. Reference answers, QA metrics, and the sealed confirmation population are absent from the generation stage; the confirmation set has been opened **zero** times. For the frozen pairwise track, an independent full replay recomputed 100,062 rows on a second GPU and reproduced every discrete selection exactly at a strict 10^{-6} tolerance; a failed post-run validator in that track is disclosed in Appendix G.

Statistics. The unit is the original question ID, and seeded selector outputs are averaged within question before any contrast is formed. Paired contrasts use a percentile bootstrap over question IDs (20,000 replicates for the main stratified comparison, 50,000 for the frozen pairwise scorer), and pre-declared F_1 contrasts within a scoring protocol are reported under a single Holm correction.[1]

Boundaries. All reported comparisons use opened or adaptive-development populations and are not independent confirmation. The external generative baseline has unknown question-level training overlap and is reported outside any leakage-free ranking. Appendix I records which failure each control diagnoses, and Appendix Table 6 the conclusion each record does and does not license.

Compute. Reader scoring dominates: each cached utility is one greedy generation (64 new tokens) with a 7B reader, batched at size 1, requiring 4–94 cached contexts per question over 6,200 training and 1,240 development instances. Selector training uses frozen BGE-M3 encodings and fits in a single GPU-day per arm; the frozen pairwise replay ran on a second GPU. No result here requires a cluster, and every table is regenerable from the released aggregate values without re-running the reader.

Ethics Statement

CompEvidenceBench is built on the public HotpotQA dataset and reproduces no private or personally identifying data. Its subject matter is evaluation methodology rather than deployment, so the release carries no model weights and no interface that could serve answers to end users. The benchmark does make a diagnostic failure visible—compressed context can drop evidence that is only jointly useful—but that is a property of existing compression methods rather than a capability we introduce. We release the failed learned methods and the negative controls alongside the positive diagnostics, so the benchmark cannot be read as a leaderboard favoring our own approach.

[1] These are the pre-registered procedures in the archived protocols. The exploratory direct-text intervals in Appendix Table 9 use 10,000 replicates and are explicitly unadjusted.

AI Use Statement

Generative AI tools assisted with literature discovery, methodological feedback and experimental design, implementation and debugging of benchmark and evaluation code, interpretation of recorded results, drafting and language editing, and preparation of illustrative figures. All AI-assisted suggestions, code, prose, and figures were reviewed against recorded experimental outputs, source materials, and cited references by the authors, who take responsibility for the final submission.

References

Ahmad Aghaebrahimian and Filip Jurčíček. Open-domain factoid question answering via knowledge graph search. In *Proceedings of the Workshop on Human-Computer Question Answering*, 2016. doi: 10.18653/v1/w16-0104. URL https://doi.org/10.18653/v1/w16-0104.

Yushi Bai, Xin Lv, Jiajie Zhang, et al. Longbench: A bilingual, multitask benchmark for long context understanding. In *ACL*, 2024. doi: 10.18653/v1/2024.acl-long.172. URL https://aclanthology.org/2024.acl-long.172/.

David Beauchemin, Richard Khoury, and Zachary Gagnon. Quebec automobile insurance question-answering with retrieval-augmented generation. In *Proceedings of the Natural Legal Language Processing Workshop 2024*, 2024. doi: 10.18653/v1/2024.nllp-1.5. URL https://doi.org/10.18653/v1/2024.nllp-1.5.

Amanda Bertsch, Eshaan Alizadeh, Tiago Pimentel, et al. Unlimiformer: Long-range transformers with unlimited length input. In *NeurIPS*, 2023. doi: 10.52202/075280-1543.

Shuyang Cao, Kaijian Zou, and Lu Wang. Sync: A synthetic long-context understanding benchmark for controlled comparisons of model capabilities. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, 2025. doi: 10.18653/v1/2025.emnlp-main.1707. URL https://doi.org/10.18653/v1/2025.emnlp-main.1707.

Jaime Carbonell and Jade Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *SIGIR*, pp. 335–336, 1998. doi: 10.1145/290941.291025.

Wenhu Chen, Hanwen Zha, Zhiyu Chen, Wenhan Xiong, Hong Wang, and William Yang Wang. Hybridqa: A dataset of multi-hop question answering over tabular and textual data. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, 2020. doi: 10.18653/v1/2020.findings-emnlp.91. URL https://doi.org/10.18653/v1/2020.findings-emnlp.91.

Hao Cheng, Hao Fang, Xiaodong Liu, and Jianfeng Gao. Task-aware specialization for efficient and robust dense retrieval for open-domain question answering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2023. doi: 10.18653/v1/2023.acl-short.159. URL https://doi.org/10.18653/v1/2023.acl-short.159.

Xuxin Cheng et al. xrag: Extreme context compression for retrieval-augmented generation with one token. *arXiv preprint arXiv:2405.13792*, 2024. doi: 10.52202/079017-3476.

Alexis Chevalier, Alexander Wettig, Aniruddha Ajith, and Danqi Chen. Adapting language models to compress contexts. In *EMNLP*, 2023. doi: 10.18653/v1/2023.emnlp-main.232.

Nadezhda Chirkova, Thibault Formal, Vassilina Nikoulina, and Stéphane Clinchant. Provence: Efficient and robust context pruning for retrieval-augmented generation. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2501.16214.

Igor Churin, Murat Apishev, Maria Tikhonova, Denis Shevelev, Aydar Bulatov, Yuri Kuratov, Sergei Averkiev, and Alena Fenogenova. Long context benchmark for the russian language. In *Proceedings of the 6th Workshop on Computational Approaches to Discourse, Context and Document-Level Inferences (CODI 2025)*, 2025. doi: 10.18653/v1/2025.codi-1.1. URL https://doi.org/10.18653/v1/2025.codi-1.1.

Giulio Corallo and Paolo Papotti. Finch: Prompt-guided key-value cache compression for large language models. *Transactions of the Association for Computational Linguistics*, 2024. doi: 10.1162/tacl_a_00716. URL https://doi.org/10.1162/tacl_a_00716.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Re. Flashattention: Fast and memory-efficient exact attention with io-awareness. In *NeurIPS*, 2022. doi: 10.52202/068431-1189.

Rajarshi Das, Ameya Godbole, Dilip Kavarthapu, Zhiyu Gong, Abhishek Singhal, Mo Yu, Xiaoxiao Guo, Tian Gao, Hamed Zamani, Manzil Zaheer, and Andrew McCallum. Multi-step entity-centric information retrieval for multi-hop question answering. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, 2019. doi: 10.18653/v1/d19-5816. URL https://doi.org/10.18653/v1/d19-5816.

Qingxiu Dong et al. Bamboo: A comprehensive benchmark for evaluating long text modeling capacities of large language models. In *LREC-COLING 2024*, 2024. doi: 10.63317/3knkq4fwny8w. URL https://aclanthology.org/2024.lrec-main.188/.

Farima Fatahi Bayat, Lechen Zhang, Sheza Munir, and Lu Wang. Factbench: A dynamic benchmark for in-the-wild language model factuality evaluation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2025. doi: 10.18653/v1/2025.acl-long.1587. URL https://doi.org/10.18653/v1/2025.acl-long.1587.

Yair Feldman and Ran El-Yaniv. Multi-hop paragraph retrieval for open-domain question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019. doi: 10.18653/v1/p19-1222. URL https://doi.org/10.18653/v1/p19-1222.

James Ferguson, Matt Gardner, and Hannaneh Hajishirzi. Iirc: A dataset of incomplete information reading comprehension questions. In *EMNLP*, 2020. doi: 10.18653/v1/2020.emnlp-main.86.

Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. Enabling large language models to generate text with citations. In *EMNLP*, 2023. doi: 10.18653/v1/2023.emnlp-main.398. URL https://aclanthology.org/2023.emnlp-main.398/.

Yichen Gong and Samuel Bowman. Ruminating reader: Reasoning with gated multi-hop attention. In *Proceedings of the Workshop on Machine Reading for Question Answering*, 2018. doi: 10.18653/v1/w18-2601. URL https://doi.org/10.18653/v1/w18-2601.

Gefei Gu, Yilun Zhao, Ruoxi Ning, Yanan Zheng, and Arman Cohan. Tail: A toolkit for automatic and realistic long-context large language model evaluation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 2024. doi: 10.18653/v1/2024.emnlp-demo.21. URL https://doi.org/10.18653/v1/2024.emnlp-demo.21.

Bernal Jimenez Gutierrez, Yu Shu, Yu Gu, and Michihiro Yasunaga. Hipporag: Neurobiologically inspired long-term memory for large language models. In *NeurIPS*, 2024. doi: 10.52202/079017-1902.

Jonathan Herzig, Thomas Müller, Syrine Krichene, and Julian Eisenschlos. Open domain question answering over tables via dense retrieval. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2021. doi: 10.18653/v1/2021.naacl-main.43. URL https://doi.org/10.18653/v1/2021.naacl-main.43.

Xanh Ho, Anh Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. In *COLING*, 2020. doi: 10.18653/v1/2020.coling-main.580.

Cheng-Yu Hsieh et al. Ruler: What's the real context size of your long-context language models? *arXiv preprint arXiv:2404.06654*, 2024.

Zhongzhan Huang, Guoming Ling, Shanshan Zhong, Hefeng Wu, and Liang Lin. Mini-longbench: The low-cost long context understanding benchmark for large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2025. doi: 10.18653/v1/2025.acl-long.560. URL https://doi.org/10.18653/v1/2025.acl-long.560.

Gautier Izacard and Edouard Grave. Leveraging passage retrieval with generative models for open domain question answering. In *EACL*, 2021. doi: 10.18653/v1/2021.eacl-main.74.

Soyeong Jeong, Jinheon Baek, Sukmin Cho, et al. Adaptive-rag: Learning to adapt retrieval-augmented large language models through question complexity. In *NAACL*, 2024. doi: 10.18653/v1/2024.naacl-long.389.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. Llmlingua: Compressing prompts for accelerated inference of large language models. In *EMNLP*, pp. 13358–13376, 2023. doi: 10.18653/v1/2023.emnlp-main.825. URL https://aclanthology.org/2023.emnlp-main.825/.

Huiqiang Jiang, Qianhui Wu, Xufang Luo, et al. Longllmlingua: Accelerating and enhancing llms in long context scenarios via prompt compression. In *ACL*, 2024. doi: 10.18653/v1/2024.acl-long.91.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *ACL*, 2017. doi: 10.18653/v1/p17-1147.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, et al. Dense passage retrieval for open-domain question answering. In *EMNLP*, 2020. doi: 10.18653/v1/2020.emnlp-main.550.

Omar Khattab and Matei Zaharia. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *SIGIR*, 2020. doi: 10.1145/3397271.3401075.

Tushar Khot, Peter Clark, Michal Guerquin, et al. Qasc: A dataset for question answering via sentence composition. In *AAAI*, 2020. doi: 10.1609/aaai.v34i05.6319.

Alex Kulesza and Ben Taskar. Determinantal point processes for machine learning. *Foundations and Trends in Machine Learning*, 5(2-3):123–286, 2012. doi: 10.1561/2200000044.

Wai-Chung Kwan, Xingshan Zeng, Yufei Wang, Yusen Sun, Liangyou Li, Yuxin Jiang, Lifeng Shang, Qun Liu, and Kam-Fai Wong. M4le: A multi-ability multi-range multi-task multi-domain long-context evaluation benchmark for large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2024. doi: 10.18653/v1/2024.acl-long.832. URL https://doi.org/10.18653/v1/2024.acl-long.832.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, et al. Natural questions: A benchmark for question answering research. *TACL*, 2019. doi: 10.1162/tacl_a_00276.

Loi Le Vu, Dong Nguyen Tien, and Tuan Dang Minh. Stratified ranking for dense passage retrieval. In *Proceedings of the 12th International Symposium on Information and Communication Technology*, 2023. doi: 10.1145/3628797.3628810. URL https://doi.org/10.1145/3628797.3628810.

Minghan Li, Ming Li, Kun Xiong, and Jimmy Lin. Multi-task dense retrieval via model uncertainty fusion for open-domain question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, 2021. doi: 10.18653/v1/2021.findings-emnlp.26. URL https://doi.org/10.18653/v1/2021.findings-emnlp.26.

Minghan Li, Xueguang Ma, and Jimmy Lin. An encoder attribution analysis for dense passage retriever in open-domain question answering. In *Proceedings of the 2nd Workshop on Trustworthy Natural Language Processing (TrustNLP 2022)*, 2022. doi: 10.18653/v1/2022.trustnlp-1.1. URL https://doi.org/10.18653/v1/2022.trustnlp-1.1.

Yingting Li. Counterfactual-augmented data for multi-hop knowledge base question answering. In *Companion Proceedings of the Web Conference 2021*, 2021. doi: 10.1145/3442442.3453706. URL https://doi.org/10.1145/3442442.3453706.

Yucheng Li, Bo Dong, Chenghua Lin, and Frank Guerin. Compressing context to enhance inference efficiency of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023. doi: 10.18653/v1/2023.emnlp-main.391. URL https://aclanthology.org/2023.emnlp-main.391/.

Zongqian Li, Yixuan Su, and Nigel Collier. 500xcompressor: Generalized prompt compression for large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2025. doi: 10.18653/v1/2025.acl-long.1219. URL https://doi.org/10.18653/v1/2025.acl-long.1219.

Hui Lin and Jeff Bilmes. A class of submodular functions for document summarization. In *ACL*, pp. 510–520, 2011. URL https://aclanthology.org/P11-1052/.

Nelson F. Liu, Kevin Lin, John Hewitt, et al. Lost in the middle: How language models use long contexts. *TACL*, 2024. doi: 10.1162/tacl_a_00638.

Ye Liu, Kazuma Hashimoto, Yingbo Zhou, Semih Yavuz, Caiming Xiong, and Philip Yu. Dense hierarchical retrieval for open-domain question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, 2021. doi: 10.18653/v1/2021.findings-emnlp.19. URL https://doi.org/10.18653/v1/2021.findings-emnlp.19.

Guangyuan Ma, Xing Wu, Zijia Lin, and Songlin Hu. Drop your decoder: Pre-training with bag-of-word prediction for dense passage retrieval. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2024. doi: 10.1145/3626772.3657792. URL https://doi.org/10.1145/3626772.3657792.

Niklas Muennighoff, Nouamane Tazi, Loic Magne, and Nils Reimers. Mteb: Massive text embedding benchmark. In *EACL*, 2023. doi: 10.18653/v1/2023.eacl-main.148.

George L. Nemhauser, Laurence A. Wolsey, and Marshall L. Fisher. An analysis of approximations for maximizing submodular set functions—I. *Mathematical Programming*, 14(1): 265–294, 1978. doi: 10.1007/BF01588971.

Jianmo Ni, Gustavo Hernandez Abrego, Noah Constant, et al. Large dual encoders are generalizable retrievers. In *EMNLP*, 2022. doi: 10.18653/v1/2022.emnlp-main.669.

Jinming Nian, Zhiyuan Peng, Qifan Wang, and Yi Fang. W-rag: Weakly supervised dense retrieval in rag for open-domain question answering. In *Proceedings of the 2025 International ACM SIGIR Conference on Innovative Concepts and Theories in Information Retrieval (ICTIR)*, 2025. doi: 10.1145/3731120.3744578. URL https://doi.org/10.1145/3731120.3744578.

Charles Packer, Sarah Fang, Shishir G. Patil, et al. Memgpt: Towards llms as operating systems. *arXiv preprint arXiv:2310.08560*, 2023.

Fabio Petroni, Aleksandra Piktus, Angela Fan, et al. Kilt: A benchmark for knowledge intensive language tasks. In *NAACL*, 2021. doi: 10.18653/v1/2021.naacl-main.200.

Jan Pfister and Andreas Hotho. Supergleber: German language understanding evaluation benchmark. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, 2024. doi: 10.18653/v1/2024.naacl-long.438. URL https://doi.org/10.18653/v1/2024.naacl-long.438.

Zexuan Qiu, Jingjing Li, Shijue Huang, Xiaoqi Jiao, Wanjun Zhong, and Irwin King. Clongeval: A chinese benchmark for evaluating long-context large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, 2024. doi: 10.18653/v1/2024.findings-emnlp.230. URL https://doi.org/10.18653/v1/2024.findings-emnlp.230.

Thilina C. Rajapakse. Dense passage retrieval: Architectures and augmentation methods. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2023. doi: 10.1145/3539618.3591796. URL https://doi.org/10.1145/3539618.3591796.

Thilina Chaturanga Rajapakse, Andrew Yates, and Maarten de Rijke. Negative sampling techniques for dense passage retrieval in a multilingual setting. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2024. doi: 10.1145/3626772.3657854. URL https://doi.org/10.1145/3626772.3657854.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. In *EMNLP*, 2016. doi: 10.18653/v1/d16-1264.

Benjamin Reichman and Larry Heck. Dense passage retrieval: Is it retrieving? In *Findings of the Association for Computational Linguistics: EMNLP 2024*, 2024. doi: 10.18653/v1/2024.findings-emnlp.791. URL https://doi.org/10.18653/v1/2024.findings-emnlp.791.

Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, et al. Colbertv2: Effective and efficient retrieval via lightweight late interaction. In *NAACL*, 2022. doi: 10.18653/v1/2022.naacl-main.272.

Kartik Sharma, Peeyush Kumar, and Yunqing Li. Og-rag: Ontology-grounded retrieval-augmented generation for large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, 2025. doi: 10.18653/v1/2025.emnlp-main.1674. URL https://doi.org/10.18653/v1/2025.emnlp-main.1674.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Musique: Multihop questions via single-hop question composition. *TACL*, 2022. doi: 10.1162/tacl_a_00475.

Jason Wei, Xuezhi Wang, Dale Schuurmans, et al. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*, 2022. doi: 10.52202/068431-1800.

Wenhan Xiong, Mo Yu, Xiaoxiao Guo, Hong Wang, Shiyu Chang, Murray Campbell, and William Yang Wang. Simple yet effective bridge reasoning for open-domain multi-hop question answering. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, 2019. doi: 10.18653/v1/d19-5806. URL https://doi.org/10.18653/v1/d19-5806.

Fangyuan Xu, Weijia Shi, and Eunsol Choi. RECOMP: Improving retrieval-augmented LMs with compression and selective augmentation. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2310.04408.

Shi-Qi Yan, Jia-Chen Gu, Yun Zhu, et al. Corrective retrieval augmented generation. *arXiv preprint arXiv:2401.15884*, 2024.

Zhilin Yang, Peng Qi, Saizheng Zhang, et al. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *EMNLP*, 2018. doi: 10.18653/v1/d18-1259.

Shunyu Yao, Dian Yu, Jeffrey Zhao, et al. Tree of thoughts: Deliberate problem solving with large language models. *arXiv preprint arXiv:2305.10601*, 2023.

Xuan Zhang et al. Infinitebench: Extending long context evaluation beyond 100k tokens. *arXiv preprint arXiv:2402.13718*, 2024. doi: 10.18653/v1/2024.acl-long.814.

Chen Zhao, Chenyan Xiong, Jordan Boyd-Graber, and Hal Daumé III. Distantly-supervised dense retrieval enables open-domain question answering without evidence annotation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021. doi: 10.18653/v1/2021.emnlp-main.756. URL https://doi.org/10.18653/v1/2021.emnlp-main.756.

Wei Zhong, Jheng-Hong Yang, Yuqing Xie, and Jimmy Lin. Evaluating token-level and passage-level dense retrieval models for math information retrieval. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, 2022. doi: 10.18653/v1/2022.findings-emnlp.78. URL https://doi.org/10.18653/v1/2022.findings-emnlp.78.

Shengyao Zhuang, Hang Li, and Guido Zuccon. Implicit feedback for dense passage retrieval. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2022. doi: 10.1145/3477495.3531994. URL https://doi.org/10.1145/3477495.3531994.

Shengyao Zhuang, Linjun Shou, and Guido Zuccon. Augmenting passage representations with query generation for enhanced cross-lingual dense retrieval. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2023. doi: 10.1145/3539618.3591952. URL https://doi.org/10.1145/3539618.3591952.

A Evaluation Units, Access Rules, and Quartet Construction

This appendix reports the archived scores, model settings, and statistical procedures behind the main results. We distinguish three experimental records: candidate-cache development, external-reader replay, and an earlier frozen pairwise confirmation. Matching budget labels alone does not justify pooling scores from these records. In particular, the three-seed text-selector ablation is the archived direct-text v3 run; it is not a three-seed measurement of the later Set-aware v2 architecture.

Table 5: Minimum reporting record for an interpretable CompEvidenceBench comparison. The record distinguishes a deployable audited result from a useful but externally trained sensitivity baseline.

Report item	**Required disclosure**	**Interpretation it protects**
Training provenance	Question-level split membership; external or synthetic training sources; all frozen checkpoints.	Prevents an answer-conditioned or overlapping training source from being treated as a fair comparison.
Configuration freeze	Architecture, loss, decoding rule, reader, tokenizer, candidate contract, and selection rule before confirmation.	Separates adaptive development from a held-out confirmation result.
Outcome reporting	All, complementary, and matched-control F1 at each budget; question counts; paired intervals.	Prevents a favorable complementary slice or duplicated budget variants from becoming the headline result.
External baseline tier	Training-overlap status and whether a result is audited or external.	Retains informative strong baselines without assigning an unsupported leakage-free rank.

Interpretation boundaries. Table 6 records the conclusion licensed by each empirical record and the conclusion it does not license. These distinctions are part of the benchmark contract, rather than post-hoc qualifications.

Table 6: Interpretation boundaries for the current CompEvidenceBench evidence. Each row states the conclusion licensed by an empirical record and the stronger conclusion that remains unsupported.

Observed record	**Supported interpretation**	**Not established**
Positive Δ quartets and candidate oracle	Budget-feasible opportunity within the enumerated slate under the construction reader.	A reader-independent semantic property or a global upper bound over all compressed texts.
Matched controls and frozen interaction tracks	The tested interaction-learning recipes do not reliably recover the diagnostic opportunity.	That interaction-aware compression cannot work.
Mistral replay with fixed retained contexts	Method differences can remain visible after changing the evaluation reader.	That the construction label or Δ is reader independent.
External BRIEF-Pro comparison	A strong external system provides a useful sensitivity baseline.	A leakage-free ranking, causal mechanism claim, or SOTA result.
HotpotQA-derived opened development	A controlled prototype for auditable analysis.	Cross-dataset prevalence or final held-out confirmation.

A.1 Budget-feasible selection and diagnostic opportunity

For a tokenizer T and a context renderer, define the feasible candidate slate as

$$\mathcal{C}_B(q, E; T) = \{c \in \mathcal{C}(q, E) : \ell_T(c) \le B\}, \qquad \widehat{c}_\theta = \arg\max_{c \in \mathcal{C}_B} s_\theta(q, c). \tag{2}$$

Here $\ell_T(c)$ counts the rendered context tokens, rather than the encoder's full question–context input. Let $F_q(c) \in [0, 1]$ denote the cached normalized answer F1. The diagnostic oracle and the mean opportunity on a set D are

$$c_q^* = \arg\max_{c \in \mathcal{C}_B} F_q(c), \qquad G_D = \frac{100}{|D|} \sum_{q \in D} \big[F_q(c_q^*) - F_q(\widehat{c}_{\mathrm{FSR}})\big]. \tag{3}$$

The oracle requires reference answers. It measures opportunity **within the enumerated slate**, not the best answer obtainable from every possible compressed text. A generative compressor can produce text outside this slate, so its score should not be interpreted as a fraction of this oracle bound.

A.2 Closed edit quartets

The implementation represents a context by retained sentence indices. For each candidate differing from the anchor by two atomic additions or removals, it forms the two corresponding singleton edits and looks up their sorted index tuples in the same candidate slate. A quartet is retained only if both singleton contexts are present. Its ordered positions are anchor, first singleton, second singleton, and joint edit. Missing singleton contexts are not assigned an invented score.

With quartet utilities u_0, u_a, u_b, u_{ab}, the audit computes

$$\Delta = u_{ab} - u_a - u_b + u_0 = (u_{ab} - u_0) - (u_a - u_0) - (u_b - u_0). \tag{4}$$

This is a discrete interaction contrast relative to an anchor. **Positive interaction does not by itself imply a correct answer**: all four utilities can be low while their contrast is positive. Nor does it establish that both singleton edits are useless. The candidate-oracle comparison is therefore reported separately from the interaction label. Matching controls on budget and candidate-count strata reduces those measured differences, but does not make the two strata identical in all aspects of question difficulty.

A.3 Training targets and inference inputs

Reference answers, cached utilities, and interaction labels may supply supervised training targets in the appropriate training partition. They are not compressor input features at inference. This distinction is necessary because the interaction learner is explicitly trained with outcome-derived supervision. At evaluation, candidate selection must finish before its chosen context is joined with the hidden audit outcomes. Original question IDs, rather than candidate rows, determine split membership and the unit of statistical resampling.

B Candidate Slate Construction

The main text fixes a candidate slate but abstracts its construction. We specify it here, because the slate determines what the diagnostic can and cannot expose.

Evidence pool. Each example is a HotpotQA question together with its supplied context, expanded sentence by sentence over the ten provided documents. A sentence inherits a document identifier and a within-document index. The pool is the full enumerated sentence set; no retrieval or relevance filter is applied at this stage, so the pool is a superset of the evidence any selector may retain.

Anchor. The anchor is not a retrieved top-k list. It is the terminal feasible state of a single rollout of the frozen additive selector that also serves as FSR: the same fixed original additivity policy is run to its final positive-feasible stopping state, and the resulting retained-sentence set becomes the anchor. Fixing the anchor this way means the quartet contrast is measured around a context that a deployable additive method actually produces, rather than around an arbitrary truncation.

Slate enumeration. A candidate is a retained-sentence set obtained from the anchor by atomic edits, where an atomic edit adds or removes one pool sentence. The released

prototype enumerates the additive direction: the anchor, every single-sentence addition, and every unordered pair of added sentences, subject to two bounds stated in the construction code—a pool bound of 4 source sentences and a slate bound of 11 candidate sets. A candidate enters the slate only if its exact rendered length is within budget. Candidates are rendered by grouping retained sentences by document, emitting a document header, and concatenating sentence texts, so that feasibility is checked on the rendered string rather than on a token estimate.

Budget feasibility. Feasibility is computed with an exact rendered-cost counter under the construction tokenizer (Qwen2), not an approximation. The counting unit is the rendered context string, and it excludes the question–context encoder input, which is why encoder-length checks and retention budgets are reported separately (Appendix C).

Quartets. For every candidate that differs from the anchor by two atomic edits, the constructor forms the two corresponding singleton edits and looks up their sorted retained-index tuples in the same slate. A quartet is retained only when both singleton contexts are present; missing singletons are not assigned an invented score. Within the released prototype the resulting slate therefore contains, in addition to the anchor, the singletons that support those quartets and the joint edits themselves.

Stratification. An example is labelled *positive-Δ complementary* when at least one closed quartet in its slate has $\Delta > 0$ under the construction reader; otherwise it is placed in the control pool. Each complementary example is matched to an unused control drawn from the same budget with the smallest candidate-count difference, with ties broken by a salted hash of the identifier pair. The released prototype contains 899 complementary and 899 matched-control training instances and 227 of each in development (Appendix Table 1 in the main text).

What the slate bounds. Because the slate is finite and enumerated around an additive anchor, the diagnostic opportunity measured against the oracle is a lower bound on the opportunity available to a constructor that searches a larger edit space. The claim licensed by the slate is therefore that *at least* the enumerated opportunity exists and was not recovered; the slate is not offered as an upper bound over all compressed texts.

C Model Objectives and Implementation Settings

The direct-text v3 ablation and Set-aware v2 share a question–candidate BGE-M3 encoder but differ in their decision architecture. Direct-text v3 applies a scalar head to each candidate representation. Set-aware v2 passes the full candidate set through two self-attention layers and adds the predicted residual to a frozen reference score. These are distinct model versions; their numeric records are kept separate below.

C.1 Utility, choice, and interaction objectives

Write the candidate scores as z_i, cached utilities as u_i, and anchor index as 0. The shared utility and choice objectives are

$$\mathcal{L}_{\text{utility}} = \frac{1}{m}\sum_{i=0}^{m-1}\big[(z_i - z_0) - (u_i - u_0)\big]^2, \tag{5}$$

$$p_i = \text{softmax}(u/\tau)_i, \quad \widehat{p}_i = \text{softmax}(z/\tau)_i, \quad \mathcal{L}_{\text{choice}} = \sum_i p_i \log \frac{p_i}{\widehat{p}_i}. \tag{6}$$

For the closed quartets $\mathcal{Q}_q$ belonging to a question, let $\widehat{\Delta}_k = z_{ab} - z_a - z_b + z_0$. The auxiliary loss and full objective are

$$\mathcal{L}_\Delta = \frac{1}{|\mathcal{Q}_q|}\sum_{k\in\mathcal{Q}_q}(\widehat{\Delta}_k - \Delta_k)^2, \qquad \mathcal{L} = \mathcal{L}_{\text{utility}} + \mathcal{L}_{\text{choice}} + \lambda\mathcal{L}_\Delta. \tag{7}$$

An empty quartet set contributes zero auxiliary loss. The utility-only arm sets $\lambda = 0$; correct-label and shuffled-label arms use $\lambda = 1$ in the documented default recipes. Shuffling permutes targets within a question, preserving their multiset while breaking their assignment to quartet positions. A permutation may leave some values unchanged, so a shuffled-label check should report the number of labels actually changed.

Table 7: Implementation settings from the archived training scripts. These are documented recipes, not a new hyperparameter sweep. Set-aware values are script defaults unless confirmed by a run record.

Setting	Direct-text v3	Set-aware v2
Candidate encoder	BGE-M3	BGE-M3
Encoder fine-tuning	Last two layers	Last two layers
Decision head	Linear scalar head	2 set-attention layers
Set attention heads	Not applicable	8
Set-layer dropout	Not applicable	0
Encoder learning rate	2×10^{-5}	2×10^{-5}
Head learning rate	10^{-3}	5×10^{-4}
Optimizer / weight decay	AdamW / 0.01	AdamW / 0.01
Gradient clipping	1.0	1.0
Choice temperature	0.1	0.1
Updates per arm	2,480	2,480 (default)
Questions per update	2	1 active + 1 inactive
Candidate encoding chunk	4 contexts	4 contexts
Maximum encoder length	8,192; no truncation	8,192; no truncation

Encoder-length checks and context budgets protect different interfaces: the former prevents silent truncation while encoding question–context pairs, and the latter constrains retained evidence supplied to the reader. In Set-aware v2, the scalar residual head is initialized to zero. The shared sampling schedule alternates questions with a nonzero interaction target and questions without one. Such balancing changes training exposure, but it does not change the evaluation population or make a development result independent.

D Complete Three-Seed Direct-Text Ablation

The following tables unpack the archived direct-text v3 scores. Each budget contains 620 opened-development instances. The table reports all three fixed training seeds, including unfavorable outcomes. These results do not establish performance of the later Set-aware v2 model.

Table 8: All seed-level answer-F1 points for direct-text v3. SD is the sample standard deviation across three training seeds, not a confidence interval.

Budget	Method	42	43	44	Mean	SD
512	FSR-additive	65.3546	65.9872	65.0684	65.4701	0.4702
512	Utility only	63.6717	64.6208	64.4821	64.2582	0.5126
512	Shuffled interaction	62.6862	64.1599	64.7637	63.8699	1.0687
512	Correct interaction	63.2022	64.0793	65.2720	64.1845	1.0389
1024	FSR-additive	64.7509	65.1088	65.1312	64.9970	0.2134
1024	Utility only	64.4142	63.4232	63.4332	63.7569	0.5693
1024	Shuffled interaction	64.2061	63.2386	63.8924	63.7790	0.4936
1024	Correct interaction	65.3304	63.3710	64.1321	64.2779	0.9878

Correct supervision exceeds the shuffled control by 0.3146 F1 at 512 tokens and 0.4988 at 1024 tokens, but both paired intervals include zero. Its differences from FSR are negative at both budgets. Thus a small positive auxiliary-label contrast should not be reported as a reliable end-to-end improvement over the frozen selector. The positive-seed count is a descriptive stability check; three fitted models are too few to characterize the full distribution of training randomness.

Table 9: Correct-interaction direct-text v3 minus each control. Intervals are conditional on the three trained seeds and are unadjusted exploratory paired intervals.

Budget	Control	Difference	95% interval	Positive seeds
512	Utility only	-0.0737	[-0.8013, +0.6609]	1/3
512	Shuffled interaction	+0.3146	[-0.4928, +1.1216]	2/3
512	FSR-additive	-1.2856	[-3.0975, +0.5311]	1/3
1024	Utility only	+0.5210	[-0.1175, +1.1552]	2/3
1024	Shuffled interaction	+0.4988	[-0.2509, +1.2862]	3/3
1024	FSR-additive	-0.7191	[-2.3118, +0.9021]	1/3

The diagnostic predictions also expose whether the interaction target has been learned on its own scale. In Table 10, predicting zero is competitive with every trained arm. Values close to the zero baseline do not establish that the target is unlearnable; they show that these fitted models have not substantially reduced its squared prediction error.

Table 10: Interaction RMSE for every direct-text v3 seed, measured on normalized F1 rather than F1 points. Lower is better; Zero predicts no interaction for every quartet.

Budget	Method	42	43	44	Zero
512	Utility only	0.319542	0.319636	0.319563	0.319501
512	Shuffled interaction	0.319618	0.319684	0.319445	0.319501
512	Correct interaction	0.319557	0.319607	0.319410	0.319501
1024	Utility only	0.275646	0.275987	0.275751	0.275683
1024	Shuffled interaction	0.275698	0.275869	0.275717	0.275683
1024	Correct interaction	0.275646	0.275981	0.275764	0.275683

E External Baselines: Full Reader Replay Results

The external-baseline scorer generates reader answers for fixed selected contexts. It is a different scoring record from candidate-cache utility evaluation. For example, internal 512-token FSR is 65.4279 under Qwen replay and 65.4701 in the direct-text candidate-cache aggregate. We retain both values with their original protocol rather than mixing their numerators or subtracting scores across records. AF-CIR is the archived residual selector in this replay study; it is not Set-aware v2.

Opened-v1 is historical development evidence. It contains 600 instances at each budget and had already been opened for analysis at the time of this replay. Its scores are not presented as untouched confirmation of the later benchmark. The internal-development population contains 620 instances at each budget. Tables 11 and 12 report both answer F1 and exact match (EM) without selecting only favorable cells.

The two readers differ in absolute score and in the magnitude of method differences. Positive gains under both readers support sensitivity of this fixed-context comparison, but do not establish reader-independent complementarity labels. Those labels were constructed under the primary reader. Unknown overlap between BRIEF-Pro training material and HotpotQA-derived questions is an additional reason to keep these results outside an audited generalization ranking.

F Actual Context Length and the Cross-Tokenizer Boundary

The cross-reader experiment preserves text, not native token counts. Contexts were constrained under the construction tokenizer and then replayed under Mistral. Retokenizing the same string can exceed the nominal 512/1024 limit. Table 13 makes this visible for both populations. "Maximum" is the maximum of the archived question-level length aggregates: for seeded methods these already average seed lengths within question, so it is not a maximum over all individual seed predictions. BRIEF-Pro has one context per question.

Table 11: Qwen fixed-context replay. FSR and AF-CIR average seeds within question; the anchor and BRIEF-Pro use a single fixed context. These populations are opened development.

Population	Budget	Method	n	F1	EM
Internal	512	Slate anchor	620	64.2107	58.3871
Internal	512	FSR	620	65.4279	59.1398
Internal	512	AF-CIR	620	65.2562	58.8710
Internal	512	BRIEF-Pro	620	78.0658	71.1290
Internal	1024	Slate anchor	620	63.3921	56.4516
Internal	1024	FSR	620	64.7823	57.2043
Internal	1024	AF-CIR	620	64.7756	57.0430
Internal	1024	BRIEF-Pro	620	73.6591	65.8065
Opened-v1	512	Slate anchor	600	63.2909	56.0000
Opened-v1	512	FSR	600	62.3292	55.0000
Opened-v1	512	AF-CIR	600	62.0680	54.8333
Opened-v1	512	BRIEF-Pro	600	74.3710	66.3333
Opened-v1	1024	Slate anchor	600	64.2070	57.3333
Opened-v1	1024	FSR	600	65.7429	58.7778
Opened-v1	1024	AF-CIR	600	65.2608	58.4444
Opened-v1	1024	BRIEF-Pro	600	77.0004	68.8333

Table 12: Mistral fixed-context replay. FSR and AF-CIR average seeds within question; the anchor and BRIEF-Pro use a single fixed context. These populations are opened development.

Population	Budget	Method	n	F1	EM
Internal	512	Slate anchor	620	59.3950	41.2903
Internal	512	FSR	620	58.4801	40.3763
Internal	512	AF-CIR	620	58.6639	40.8602
Internal	512	BRIEF-Pro	620	66.2776	49.6774
Internal	1024	Slate anchor	620	57.9916	39.5161
Internal	1024	FSR	620	57.5309	38.8172
Internal	1024	AF-CIR	620	56.9152	38.3871
Internal	1024	BRIEF-Pro	620	66.8270	49.6774
Opened-v1	512	Slate anchor	600	57.6205	39.3333
Opened-v1	512	FSR	600	58.6767	40.6111
Opened-v1	512	AF-CIR	600	58.6415	40.9444
Opened-v1	512	BRIEF-Pro	600	67.5639	52.3333
Opened-v1	1024	Slate anchor	600	57.3958	40.6667
Opened-v1	1024	FSR	600	56.9583	39.9444
Opened-v1	1024	AF-CIR	600	56.5986	39.7778
Opened-v1	1024	BRIEF-Pro	600	65.6956	50.5000

Table 13: Native context token counts in each replay reader. Nominal budgets refer to the construction tokenizer; Mistral replay is not a native hard-cap comparison.

Tokenizer	Population	Budget	Method	Mean	Maximum
Qwen	Internal	512	FSR	395.37	512.00
Qwen	Internal	512	AF-CIR	395.28	512.00
Qwen	Internal	512	BRIEF-Pro	207.84	495.00
Qwen	Internal	1024	FSR	463.20	1022.00
Qwen	Internal	1024	AF-CIR	471.20	1022.00
Qwen	Internal	1024	BRIEF-Pro	206.45	491.00
Qwen	Opened-v1	512	FSR	395.43	512.00
Qwen	Opened-v1	512	AF-CIR	398.28	512.00
Qwen	Opened-v1	512	BRIEF-Pro	210.81	512.00
Qwen	Opened-v1	1024	FSR	461.45	1020.00
Qwen	Opened-v1	1024	AF-CIR	467.56	1020.67
Qwen	Opened-v1	1024	BRIEF-Pro	207.36	573.00
Mistral	Internal	512	FSR	459.82	638.33
Mistral	Internal	512	AF-CIR	459.67	638.33
Mistral	Internal	512	BRIEF-Pro	226.89	557.00
Mistral	Internal	1024	FSR	539.22	1202.00
Mistral	Internal	1024	AF-CIR	548.39	1202.00
Mistral	Internal	1024	BRIEF-Pro	225.81	520.00
Mistral	Opened-v1	512	FSR	461.22	636.00
Mistral	Opened-v1	512	AF-CIR	464.38	620.00
Mistral	Opened-v1	512	BRIEF-Pro	230.14	551.00
Mistral	Opened-v1	1024	FSR	536.84	1214.67
Mistral	Opened-v1	1024	AF-CIR	543.81	1214.67
Mistral	Opened-v1	1024	BRIEF-Pro	226.91	626.00

For example, internal 512-token BRIEF-Pro contexts have mean length 207.84 in Qwen tokens and 226.89 in Mistral tokens, with a Mistral maximum of 557. The corresponding seeded FSR aggregate reaches 638.33 Mistral tokens. A claim of equal *native* Mistral budgets would therefore be unsupported. Reapplying a Mistral cap would create different contexts and require a new evaluation; no such result is inferred here.

Shorter retained contexts also do not prove a faster complete system. BRIEF-Pro generates a compressed context before the reader runs, while selector latency was not measured comparably for every baseline. Shared reader batches further prevent assigning their total runtime to one method. We therefore report context lengths without claiming a latency or throughput advantage.

G Frozen Pairwise Track: Complete Scale Comparison

This earlier experiment trained at the 6,200-question scale and made predictions on 1,200 previously sealed confirmation instances, balanced across the two budgets. It used a different candidate-edit pipeline and is not confirmation of either direct-text v3 or Set-aware v2. The archived summary contains small- and large-scale models at seeds 42, 43, and 44. Table 14 reports every method at both scales; means and sample SDs below are computed from the stored seed-level scores.

The frozen study's stronger success criterion did not pass. At large scale, decision-pairwise minus atomic additive was -0.1107 F1 at 512 tokens and -0.5534 at 1024; its differences from the matched additive correction were $+0.5483$ and -1.2723. These mixed contrasts are inconsistent with dependable improvement across budgets.

The archived independent report also found a 512-token scale interaction of $+1.7362$ F1 relative to the matched additive correction, with two-sided interval $[0.0172, 3.5405]$. This difference-in-differences is a narrower observation than large-scale superiority: the direct 512-token gain over that correction was only 0.5483, and the overall success criterion failed. The 433-question content-disjoint auxiliary slice is descriptive and does not replace the full confirmation population.

Table 14: Earlier frozen pairwise track: complete F1 scores, 600 confirmation instances per budget. These are distinct candidates and models from the main development experiment.

Scale	Budget	Method	42	43	44	Mean	SD
small	512	Atomic additive	62.069	62.204	63.168	62.480	0.599
small	512	Exact-pair MSE	62.997	61.160	59.372	61.176	1.813
small	512	Decision pairwise	61.178	62.358	62.026	61.854	0.609
small	512	Additive correction	63.112	63.423	62.590	63.042	0.421
small	1024	Atomic additive	63.744	64.936	64.687	64.456	0.629
small	1024	Exact-pair MSE	61.695	62.360	63.178	62.411	0.743
small	1024	Decision pairwise	63.988	64.405	64.594	64.329	0.310
small	1024	Additive correction	63.765	62.987	64.725	63.826	0.871
large	512	Atomic additive	62.856	63.364	63.142	63.121	0.255
large	512	Exact-pair MSE	63.545	63.824	62.364	63.244	0.775
large	512	Decision pairwise	62.777	62.566	63.686	63.010	0.595
large	512	Additive correction	62.622	62.051	62.712	62.462	0.358
large	1024	Atomic additive	64.593	65.793	65.659	65.348	0.658
large	1024	Exact-pair MSE	64.093	65.322	64.452	64.622	0.632
large	1024	Decision pairwise	65.313	64.713	64.359	64.795	0.483
large	1024	Additive correction	65.551	66.424	66.226	66.067	0.458

G.1 Mechanical validation recovery

The original postrun validator failed after predictions were complete because it compared integer budget keys with their JSON string forms in a key-order-sensitive digest. A separate recovery validator normalized keys only for the expected training-info schema. The stored protocol, models, predictions, and statistical logic remained unchanged. This is a validation-recovery disclosure, not an additional experimental run or an opportunity to tune against confirmation outcomes.

H Submission Protocol and Reporting Record

Construction audit, model development, and final comparison require separate access rules. A method may choose its architecture, loss, and decoding configuration using public training inputs and compressor-visible development fields, but not reference answers, candidate F1, oracle choices, or interaction labels. At inference, a selection must depend only on the question, evidence pool, candidate slate, and budget; answer access would change the task to answer-conditioned ranking.

After the development configuration is frozen, an audited submission should use a question-disjoint confirmation population with the same reader, tokenization, candidate contract, and pre-declared selection rule. The reported unit is the **original question**, not a budget variant or candidate. Stochastic seed outputs should be aggregated within question before paired resampling. Each result should report All, complementary, and matched-control strata at both budgets with their question counts.

External compressors with incomplete pretraining, synthetic-data, or question-level provenance can be replayed through the fixed readers and reported as sensitivity baselines. They cannot establish a leakage-free leaderboard rank or turn a subset advantage into a general compression claim. Reports should state which comparison each score supports and its access conditions. Table 5 states the minimum disclosure record and Table 6 the interpretation each record licenses.

I Paired Statistics and Interpretation of Failures

For methods A and B and fixed trained seeds $s = 1, \ldots, S$, define a question-level paired difference in F1 points as

$$d_q = \frac{100}{S} \sum_{s=1}^{S} \left[F_q(c_{A,s}) - F_q(c_{B,s}) \right], \qquad \widehat{\delta} = \frac{1}{N} \sum_{q=1}^{N} d_q. \tag{8}$$

For a single-context external method, its score is paired with the within-question seed mean of the selector. A bootstrap replicate samples N question IDs with replacement and recomputes the paired mean. The reported percentile interval is

$$\mathrm{CI}_{95} = \left[Q_{0.025}(\widehat{\delta}^{*(1)}, \ldots, \widehat{\delta}^{*(R)}), Q_{0.975}(\widehat{\delta}^{*(1)}, \ldots, \widehat{\delta}^{*(R)}) \right]. \tag{9}$$

Candidate rows and seed predictions are not treated as independent questions. The main stratified comparison uses 20,000 replicates; the archived direct-text intervals in Table 9 used 10,000, and the earlier frozen pairwise report used 50,000 independent question-level replicates. We preserve each analysis's original procedure. Bootstrap intervals conditional on fixed models do not account for all randomness in retraining, nor do they correct adaptive reuse of a development population. Table 9 explicitly retains its unadjusted exploratory status.

I.1 Which failure does each control diagnose?

An oracle gap diagnoses **available candidate opportunity**. A correct-versus-shuffled contrast diagnoses whether the intended assignment of interaction targets improves the chosen learning recipe. A learned-versus-FSR contrast diagnoses whether any such improvement survives a strong frozen selector. Cross-reader replay diagnoses whether a fixed-context method difference survives a reader change. None of these checks alone establishes that the compressor has learned a general mechanism of complementary evidence retention.

The RMSE results in Table 10 and the paired intervals in Table 9 answer different questions. Accurate interaction regression need not flip a candidate argmax, and a flipped argmax need not improve answer F1. Conversely, a utility-trained selector can improve choices without reconstructing the interaction contrast. We report both target-level error and downstream choices to distinguish these cases.

I.2 Evidence that remains unavailable

The appendix adds detail from completed records; it does not add an independent confirmation population for the current set-aware model, an audited training-overlap guarantee for BRIEF-Pro, or reconstructed complementarity labels under a second reader. The opened-v1 replay, the frozen earlier pairwise track, and the current candidate-cache development results must retain their distinct roles. Each record can then be assessed under its own split, model version, and scoring procedure.

J Scope of the Test

CompEvidenceBench evaluates *selection under a finite candidate slate*. It does not evaluate whether a compressor can synthesize arbitrary new evidence, retrieve a missing document, or improve a reader's reasoning in the absence of sufficient evidence. The constrained slate is intentional: it makes the counterfactual question—whether a budget-feasible jointly useful context was retained—testable. A generative compressor is permitted to produce a new compressed text, but its output is still judged under the same reader and budget contract.

The benchmark also does not turn Δ into a test-time target. The interaction score is computed only after fixed-reader outcomes and reference answers are available. This distinction is needed for a fair interpretation of the oracle: the oracle diagnoses an opportunity in the slate, whereas a real compressor must infer useful combinations from question and evidence alone. Appendix Table 5 states the minimum disclosure record and Table 6 the interpretation each record licenses.

K Mechanism Hypotheses for the Weak Interaction Signal

The main text states three hypotheses that bound how much an explicit interaction term can contribute on the released data. We develop each one and state what would falsify it.

Target sparsity. At 512 tokens, approximately 87.8% of question-weighted exact-Δ targets are zero. A zero-heavy target distribution has two consequences: the effective supervision for an interaction head is much smaller than the nominal quartet count, and a small positive shrinkage of a correction term cannot reduce squared error below the zero predictor. Table 10 is consistent with this: predicting zero is competitive with every trained arm on nine independent units, and across six seed–split combinations the fitted expectation $E[p^*\Delta]$ is negative while outputs never reach their ± 0.20 saturation bound. *Falsification:* an interaction objective that reweights toward non-tied quartets, or that conditions on candidate pairs with distinct reader F1, should reduce target-level error below the zero predictor. If it does not, target sparsity is not the binding constraint.

Representation absorption. The candidate encodings are question-conditioned and include global context, so scoring is additive at the *scoring* level but not at the representation level: a sum of per-candidate scores over globally conditioned encodings is not equivalent to independent scoring. If the encoder has already absorbed cross-candidate information, an explicit pairwise term is asked to model what remains. This is consistent with an interaction head that is poor at Δ regression while the underlying selector stays strong. *Falsification:* swapping in per-candidate encodings computed without global context, and re-running the same interaction objective at matched scale, should change the matched-control contrast if absorption is the cause.

Decision granularity. Selection is an argmax over a slate with a narrow utility spread, and most candidate position changes are ties in normalized reader F1. Corrections therefore change the discrete selection far more often than they change the answer, which decouples target-level error from downstream F1. The shuffled-label control is the direct evidence: permuting Δ within question changed labels for 1,413 of 1,517 questions with active quartet supervision, yet for seed 42 the correct and shuffled variants selected the same candidate on all 1,240 development questions. *Falsification:* evaluating on a stratum where the utility spread is wide relative to the correction scale should restore a dependence of answer F1 on the interaction targets.

These hypotheses are separable rather than mutually exclusive. Each yields a distinct intervention, and the released labels, candidate slates, and cached utilities make each intervention runnable without new annotation.