From Evidence to Exact Answers: Copy-Preserving Compression and Conversion-Aware Calibration for Frozen SLM RAG

Anonymous ACL submission

Abstract

Frozen small language model (SLM) readers in retrieval-augmented generation (RAG) can fail even after retrieval has found answer-bearing evidence: on five LongBench tasks with LFM2.5-Instruct 1.2B, Raw RAG with its default prompt converts only 43% of answer-bearing predictions into exact matches. We treat frozen SLM RAG as a two-stage problem—copy-preserving evidence compression and conversion-aware deployment calibration—and instantiate each stage with one system. PEC-HOP is a non-parametric pointer-preserved evidence compressor; the Minimal Copy-preserving Evidence Compass (MCE-Compass) is a dev-time calibrator that selects an (interface, prompt) pair per (reader, task) cell. In a matched-prompt head-to-head on three multi-hop LongBench tasks $\times$ three frozen readers, PEC-HOP is Pareto-best among compressors, with $+4.28$ pp macro EM over LLMLingua-2 and $+1.61$ pp over Provence at 2.8–4.8$\times$ faster median latency and no trained compressor on the critical path. Across five tasks, MCE-Compass improves macro EM over Raw RAG by $+9.36$, $+15.39$, and $+2.39$ pp on LFM2.5-Instruct 1.2B, Qwen3-4B, and Gemma-4-E4B at $2.02\times$, $5.69\times$, and $1.34\times$ latency speedups, and beats the single globally dev-selected policy. A 20-partition dev-size sweep and a 66,581-article HotpotQA retrieval stress test indicate the gains are not single-split or bundled-context artifacts.

1 Introduction

When latency, memory, and hardware budgets are tight, small language models (SLMs) in the 1B–4B range have become attractive low-cost readers for knowledge-intensive question answering (Amini et al., 2025; Yang et al., 2025). Retrieval-augmented generation (RAG) is a natural fit for the setting because a small frozen reader can be paired with external evidence instead of being retrained for every knowledge update (Lewis et al., 2020). The usual starting point is simple: retrieve several raw passages, concatenate the passages, and ask the reader for a short answer.

A practical failure mode appears when a frozen SLM reader is given several concatenated retrieved passages. Longer raw prompts do not only increase prefill and key-value cache cost; such prompts can also make an SLM less disciplined about copying the answer span (Jiang et al., 2023, 2024). In the LongBench setting, Raw RAG often emits a sentence that contains the gold answer but is not formatted as the exact answer. On five tasks, only 43% of Raw RAG's answer-bearing predictions become exact matches. The missing step is therefore evidence-to-answer conversion: a frozen reader must see evidence in a copy-preserving form that makes the metric-compatible answer cheap to emit. This is a deployment issue for frozen SLM RAG, not only a retrieval-recall or context-compression issue.

Existing compression methods address part of the issue but leave two practical gaps for frozen SLM readers. Query-time compressors (Xu et al., 2023; Jiang et al., 2023, 2024; Yu et al., 2024; Chirkova et al., 2025) reduce context length, but add another model pass on the critical path. Offline summaries such as Recursive Abstractive Processing for Tree-Organized Retrieval (RAPTOR) (Sarthi et al., 2024) can be compact, but may paraphrase or omit names, dates, and numbers that a question-answering (QA) reader must copy. Thus the input interface should be short, retrievable, and faithful to source strings; it must also be chosen conservatively, because the best interface can change with the reader and task.

We address these two gaps with a method and a framework. PEC-HOP (pointer-preserved evidence-card hopping) is a non-parametric multi-hop compressor that represents evidence as com-

pact source-faithful cards—title, anchor entities, a verbatim source sentence, and a back-pointer to the passage—retrieved in two stages (anchor-overlap–reranked seeds, then anchor-expanded hop-2) and hydrated into a local source window. No trained compressor sits on the critical path.

PEC-HOP alone is not universally best: sentence-only retrieval suits single-document QA, and some (reader, task) cells prefer Raw RAG as a safe fallback. The Minimal Copy-preserving Evidence Compass (MCE-Compass) is a deployment-time calibrator over the fixed copy-preserving family {Raw RAG, Sentence-only, PEC-HOP} crossed with five decoding disciplines. From a small dev split it keeps accuracy-non-inferior candidates and picks the lowest-burden point on the EM/F1–latency–context frontier, with conversion as the tie-breaker; no reader training or new retriever is required.

The contributions are:

1. **A two-bottleneck diagnostic for frozen SLM RAG.** Failure is not explained by retrieval recall alone: evidence-side reader burden and output-side conversion fail independently. On LFM2.5-Instruct 1.2B, Raw RAG with the default prompt converts only 43% of answer-bearing predictions into exact matches, while a copy-oriented few-shot discipline raises this to 88–93%. Prior compression work targets only the evidence side and is evaluated mostly on $\geq$ 7B readers, where conversion is not binding.

2. **PEC-HOP, a non-parametric pointer-preserved compressor.** In a matched-prompt head-to-head on three multi-hop LongBench tasks $\times$ three frozen SLM readers, PEC-HOP achieves higher macro EM than LLMLingua-2 (+4.28 pp) and Provence (+1.61 pp) at 2.8–4.8$\times$ faster median latency, with no trained compressor on the critical path.

3. **MCE-Compass, a deployment-time calibrator.** Over the fixed family {Raw RAG, Sentence-only, PEC-HOP} crossed with five decoding disciplines, the rule keeps accuracy-non-inferior candidates and picks the lowest-burden point. It improves macro EM over Raw RAG by +9.36, +15.39, and +2.39 pp on LFM2.5, Qwen3-4B, and Gemma-4-E4B at 2.02$\times$, 5.69$\times$, and 1.34$\times$ latency speedups, and beats the single globally dev-selected policy.

4. **Robustness controls.** A 20-partition dev-size sweep and a 66,581-article HotpotQA external-retrieval stress test indicate the gains are not single-split or bundled-context artifacts.

2 Related Work

RAG, dense retrieval, and multi-hop QA. Retrieval-augmented generation (RAG) (Lewis et al., 2020) pairs a generator with retrieved evidence. Dense Passage Retrieval (Karpukhin et al., 2020) made the recipe practical for open-domain readers, and multi-hop retrievers extend it to questions whose answer depends on evidence from multiple passages (Xiong et al., 2020); HotpotQA exemplifies the multi-hop pressure point, requiring not only answer-bearing text but also connections across documents (Yang et al., 2018). Retrieval-focused work therefore improves *which* passages are retrieved. In standard RAG, however, the reader is still commonly given raw passage text, even though long-context readers may fail to use relevant evidence uniformly across a prompt (Liu et al., 2024). Fusion-in-Decoder and Atlas reduce reader burden by separately encoding passages or jointly training retrieval and reading components (Izacard and Grave, 2021; Izacard et al., 2023), but require architectural or training changes. The setting considered here is more constrained: the reader remains frozen, and the only available intervention is the textual evidence interface placed in front of it.

Compression, pruning, and generated notes. Compression reduces prompt burden, but the mechanisms differ in stage, supervision, and faithfulness. LLMLingua, LongLLMLingua, and LLMLingua-2 compress prompts at query time using task-agnostic compression models (Jiang et al., 2023, 2024; Pan et al., 2024). ReComp and Provence select or rewrite shorter evidence for RAG (Xu et al., 2023; Chirkova et al., 2025); AttentionRAG uses reader attention and therefore adds an additional reader-side forward pass (Fang et al., 2025). Chain-of-Note inserts note-style intermediate evidence (Yu et al., 2024), RAPTOR builds offline abstractive summaries (Sarthi et al., 2024), and PISCO and xRAG distill documents or replace text with learned bridge tokens (Louis et al., 2025; Cheng et al., 2024). Shorter evidence can help RAG, but these mechanisms create deployment issues for frozen SLM

readers: query-time compressors add a model pass on the critical path, generated notes can paraphrase names/dates/numbers that QA readers must copy, and learned representations break source faithfulness. Existing evaluations also focus on $\geq$ 7B readers, where output-side conversion is not the binding constraint. In contrast, MCE family keeps reader-visible text plain and source-faithful; PEC-HOP occupies a favorable accuracy–latency Pareto region among practical compressors (Table 1), and Appendix Table 5 summarizes the closest design axes.

Small readers and the remaining deployment gap. Sub-2B/4B SLM families such as LFM2 (Amini et al., 2025) and Qwen3 (Yang et al., 2025) make low-cost RAG deployment attractive, and LongBench (Bai et al., 2024) provides a fixed-context testbed. Adjacent threads — adaptive routing and corrective RAG, selective QA (Jeong et al., 2024; Yan et al., 2024; Kamath et al., 2020; Kadavath et al., 2022), and constrained decoding (Hokamp and Liu, 2017; Post and Vilar, 2018) — calibrate high-level routing, abstention, or lexical form, but do not decide which reader-visible evidence form should be used once a frozen reader must answer. The remaining gap for frozen SLM RAG is therefore two-sided: compression targets the evidence side but can add critical-path calls or paraphrase copy-critical strings, while output-side calibration is left underspecified. MCE-Compass keeps the reader fixed and calibrates the evidence interface and decoding discipline together, retaining Raw RAG as a fallback when compression is not dev-safe.

3 Method

The method has two stages: (i) compress retrieved evidence into a copy-preserving form (Sections 3.1–3.2), and (ii) treat decoding as a separate calibration axis and let MCE-Compass select the lowest-burden dev-safe (interface, prompt) per reader–task cell (Section 3.3). Figure 1 shows the flow.

3.1 Setting and Copy-Preserving Evidence Interfaces

A LongBench (Bai et al., 2024) instance is $(q, C, \mathcal{A})$: a question, a long context (5–130k characters), and reference answers. For multi-document tasks C contains $|\mathcal{P}|$ titled passages $\mathcal{P} = \{(t_i, b_i)\}_{i=1}^{|\mathcal{P}|}$; for single-document tasks

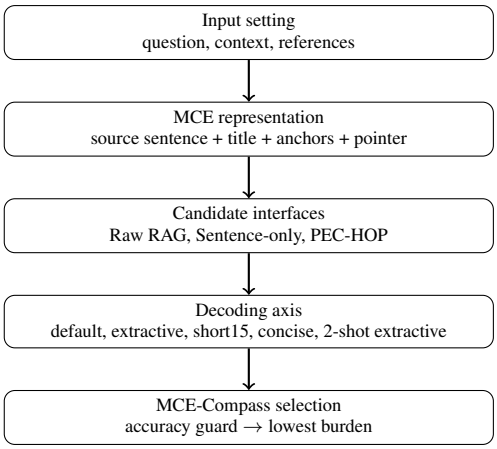

Figure 1: Method flow. The card representation defines what can be retrieved; the interface and decoding axes define the candidate grid; the conversion-aware calibration rule selects the cheapest candidate that preserves development-set accuracy.

$|\mathcal{P}|$=1. A reader $\mathcal{M}$ produces $\hat{y} = \mathcal{M}(q, c)$ from a compiled prompt c; the goal is to construct c so as to preserve QA quality while reducing $|c|$, latency, and any auxiliary inference cost. The interface space is fixed before calibration: Raw RAG exposes raw passages, Sentence-only exposes directly retrieved source sentences, and PEC-HOP exposes pointer-preserved bridge cards. A design-axis comparison against the closest compression baselines is deferred to Appendix A.

Conversion is measured separately from answer presence. Let $\mathrm{loose}(\hat{y}, \mathcal{A})$=1 when the prediction contains an acceptable answer string after normalization, and let $\mathrm{EM}(\hat{y}, \mathcal{A})$ be exact match. For candidate c, we define

$$\gamma(c) = P(\mathrm{EM} = 1 \mid \mathrm{loose} = 1),$$

estimated on the dev split. High conversion means the reader not only sees the answer, but emits it in the required format.

For each passage (t, b), b is tokenized into sentences, with one card emitted per sentence as a 4-line record with a source pointer (PTR) field:

```
TITLE: t    ANCHORS: a1,a2,...
FACT: sj    PTR: p=i;s=j
```

ANCHORS are lightweight lexical cues extracted by regex; implementation details and caps are in Appendix D. Cards are embedded once with frozen gte-small (Li et al., 2023) into a per-question Facebook AI Similarity Search (FAISS) index. FACT preserves the source span, and PTR supports source-window hydration at read time.

3.2 Evidence Compressors

The candidate interfaces correspond to distinct failure modes. Raw RAG is the safe fallback when compression is not dev-safe. Sentence-only is cheap and copy-preserving but often too local for multi-hop questions, where a matched sentence may name only a bridge entity. PEC-HOP targets this case by keeping source strings verbatim, adding lightweight bridge structure, and hydrating hits back into local evidence windows.

Raw RAG (baseline). The top-k raw passages are retrieved $\mathcal{P}_q \subseteq \mathcal{P}$ by question–passage cosine similarity; the first 200 tokens of each passage are concatenated as c. The retrieval budget is $k = 5$.

Sentence-only (baseline). The top-k source sentences are retrieved through the same FAISS index, without card schema, second-hop retrieval, or hydration. This isolates the contribution of the card schema and hop machinery.

PEC-HOP. PEC-HOP implements the bridge-aware interface without a trained compressor, additional large language model (LLM) call, or reader modification. As an *evidence compressor* it (i) reranks a seed pool by question similarity plus anchor and wh-type overlap, (ii) expands the query with seed anchors and titles for a second retrieval pass, and (iii) hydrates selected cards through PTRs into local source windows. Decoding is not part of the compressor; the prescribed conversion discipline for PEC-HOP is specified separately in Section 3.3 as one point on the decoding axis, so that prompt and interface effects can be ablated independently. Prompt and ablation details are in Appendices E and B; Algorithm 1 gives the full retrieval procedure.

3.3 Compass selector: Conversion-Aware Calibration

The candidate family is fixed before looking at test data. Raw RAG and Sentence-only are each crossed with five decoding disciplines—default, extractive with a 5-word cap, short15 with a 15-word cap, concise, and 2-shot extractive—and PEC-HOP contributes its prescribed 2-shot extractive configuration, giving 11 candidates per (reader, task) cell. Decoding is thus a separate calibration axis from the evidence interface, and the full prompt templates are in Appendix E. The Compass selector is not a learned router; it chooses one dev-selected candidate per cell and keeps it fixed for all test queries in that cell.

Algorithm 1 PEC-HOP: anchor-overlap rerank + anchor-expanded hop retrieval

Require: Question q, card index $\mathcal{I}$, top-k
$\mathcal{S}_0 \leftarrow \mathrm{topk}(\mathcal{I}, q, 2k)$ // seed pool
$A_q \leftarrow \mathrm{QuestionAnchors}(q)$; $T_q \leftarrow \mathrm{whType}(q)$
for $c \in \mathcal{S}_0$ **do**
  $r(c) \leftarrow |\mathrm{ANCHORS}(c) \cap A_q|$
  $\tilde{s}(c) \leftarrow \cos(q, c) + 0.1 r(c) + 0.05 \mathbb{1}[T_q \in c]$
end for
$\mathcal{S} \leftarrow \mathrm{topk}_{\tilde{s}}(\mathcal{S}_0, k)$ // reranked seeds
$A \leftarrow \emptyset$
for $c \in \mathcal{S}_{1:3}$ **do**
  $A \leftarrow A \cup \mathrm{ANCHORS}(c) \cup \{\mathrm{TITLE}(c)\}$
end for
$A \leftarrow \mathrm{dedup}(A)_{1:8}$
$q' \leftarrow q \,\|\, \mathrm{join}(A)$
$\mathcal{R} \leftarrow \mathrm{topk}(\mathcal{I}, q', 2k) \setminus \mathcal{S}$
$\mathcal{H} \leftarrow \mathcal{R}_{1:k}$ // hop-2 cards
$m \leftarrow \max(2, k - |\mathcal{H}|)$
$\mathcal{F} \leftarrow \mathcal{S}_{1:m} \,\|\, \mathcal{H}$
$\mathcal{F} \leftarrow \mathcal{F}_{1:k+2}$
return $\mathrm{hydrate}(\mathcal{F})$ // ± 1-sentence window

Deployment proposition. For frozen-reader RAG, Compass selector is an accuracy-safe operating-point selector: keep dev-non-inferior candidates, discard dominated points, then choose the lowest reader burden, using conversion as a near-tie breaker. Admissibility uses a one-dev-example EM margin (or its paired-bootstrap lower bound) with a 3 pp F1 guard; the F1 guard prevents short answers from winning long-form QA purely by being cheap. Reader burden is

$$B(c) = \log \frac{\ell(c)}{\ell(RawRAG)} + 0.35 \log \frac{u(c)}{u(RawRAG)},$$

where the logarithms compare multiplicative changes from Raw RAG and the fixed 0.35 coefficient keeps latency as the primary burden while accounting for context tokens (Appendix C reports a sensitivity check). Among non-dominated admissible candidates within a 0.10 log-burden band of the lightest, the selector prefers higher dev conversion $P(\mathrm{EM} \mid \mathrm{loose})$; Algorithm 2 gives the full procedure.

4 Experiments

Evaluation regime and datasets. The main evaluation isolates interface–discipline selection within the fixed family from Section 3. LongBench (Bai et al., 2024) supplies shared source

Algorithm 2 Compass selector calibration for one reader–task cell

Require: Development set D; fixed candidate family $\mathcal{C}$; Raw RAG reference c_0
Evaluate each $c \in \mathcal{C}$ on D: EM $a(c)$, F1 $f(c)$, latency $\ell(c)$, tokens $u(c)$, conversion $\gamma(c)$
$c^\star \leftarrow \arg\max_{c \in \mathcal{C}} (a(c), f(c), -\ell(c))$
$\epsilon \leftarrow 1/|D|$ // one dev example
$\mathcal{A} \leftarrow \emptyset$
for $c \in \mathcal{C}$ **do**
  $L_c \leftarrow$ paired-bootstrap lower bound of $a(c) - a(c^\star)$
  if $a(c) \geq a(c^\star) - \epsilon$ or $L_c \geq -\epsilon$ **then**
    if $f(c) \geq f(c^\star) - 3$ pp **then**
      $\mathcal{A} \leftarrow \mathcal{A} \cup \{c\}$
    end if
  end if
end for
$B(c) \leftarrow \log(\ell(c)/\ell(c_0)) + 0.35 \log(u(c)/u(c_0))$
$\mathcal{P} \leftarrow$ nondominated points in $\mathcal{A}$ under high EM, low B, high γ
$B_{\min} \leftarrow \min_{c \in \mathcal{P}} B(c)$
$\mathcal{P}_{0.10} \leftarrow \{c \in \mathcal{P} : B(c) \leq B_{\min} + 0.10\}$

return $\arg\max_{c \in \mathcal{P}_{0.10}} (\gamma(c), f(c), -u(c))$

contexts for five English tasks: HotpotQA (Yang et al., 2018), 2WikiMultiHopQA (Ho et al., 2020), and MuSiQue test multi-hop bridge evidence, while Qasper and multifieldqa_en stress answer formatting. We evaluate three frozen readers, add a 20-partition dev-size sweep, and include a 66,581-article HotpotQA external-retrieval stress test.

Reader models and controlled variables. All comparisons use frozen SLM readers. LFM2.5-Instruct 1.2B (Amini et al., 2025) is the main reader; Qwen3-4B (Yang et al., 2025) and Gemma-4-E4B-it (Gemma Team and Google DeepMind, 2026) test whether the policy transfers across SLM families. Reader weights, Q4_K_M GPT-Generated Unified Format (GGUF) quantization, `llama.cpp` runtime, context limit (`n_ctx=2048`), retrieval encoder (frozen gte-small), top-k retrieval budget, question order, greedy decoding, and hardware are held fixed within each comparison. Only the evidence interface and decoding discipline are varied.

Baselines. Raw RAG is the uncompressed safe fallback. Sentence-only is the copy-preserving minimal-interface baseline: it removes card schema, bridge expansion, and hydration while still exposing source strings. LLMLingua-2 (Pan et al., 2024) and Provence (Chirkova et al., 2025) are the closest trained query-time compressor baselines because these methods compress evidence before the same frozen reader without retraining the reader. Both compressors are run at their default settings on the same RTX 3090 GPU as the reader, and the latency reported in Table 1 and Table 7 includes the compressor forward pass plus reader prefill and decode. Head-to-head compressor comparisons hold the prescribed 2-shot extractive prompt fixed so that Table 1 isolates the evidence interface.

Evaluation metrics and comparison protocol. Accuracy is measured with Exact Match (EM) and token-overlap F1 score. Efficiency is measured in two ways: end-to-end latency from question submission to answer emission, and reader-input context tokens after truncation to 1500 tokens. As a diagnostic for evidence-to-answer conversion, loose match records whether the prediction contains an acceptable answer string after normalization, and conversion is $P(\text{EM} \mid \text{loose})$. Significance is assessed with a paired bootstrap using 10,000 resamples paired by question index. Non-inferiority uses a pre-specified -3 pp margin and reports the one-sided 95% lower bound for the difference between PEC-HOP and Raw RAG.

4.1 Main Results

The candidate family per (reader, task) cell is fixed before calibration: Raw RAG and Sentence-only are each crossed with five decoding disciplines, and PEC-HOP contributes one prescribed conversion-aware configuration. The Compass selector is calibrated on the first 50 examples per task (40 for multifieldqa_en); the chosen interface/discipline pair is evaluated on the held-out test split. The headline numbers are in Table 2. Supporting grids and diagnostic controls are reported after the main results.

PEC-HOP is the Pareto-best compressor. Holding the 2-shot extractive prompt fixed, this head-to-head isolates the effect of the evidence interface: PEC-HOP is Pareto-best among compressors, achieving the highest macro EM, lower median latency than the trained baselines, and no trained compressor on the critical path (Table 1). The compressor ranking is robust to the prompt itself: under a matched 0-shot extractive template, PEC-HOP remains the best method on LFM2.5 (Appendix Table 9), so the Table 1 advantage is not prompt-conditional on the main reader. On Gemma-4-E4B and Qwen3-4B the prompt becomes the dominant lever, which we separate out

Method	LFM2.5-1.2B		Gemma-4-E4B		Qwen3-4B	
	EM	lat (ms)	EM	lat (ms)	EM	lat (ms)
Raw RAG (no compression, 2-shot)	20.17	251	24.00	884	34.17	597
Sentence-only	17.67	241	15.33	486	22.33	316
LLMLingua-2 (Pan et al., 2024)	18.83	1019	18.67	1281	26.67	867
Provence (Chirkova et al., 2025)	21.17	654	21.67	644	29.33	540
PEC-HOP (ours)	**23.17**	**138**	**23.33**	**267**	**30.50**	**261**

Table 1: Compressor head-to-head on three multi-hop LongBench tasks and three frozen SLM readers, with the 2-shot extractive prompt held fixed. Bold marks the best compressor-block value. Raw RAG is included as an uncompressed reference, not a compressor baseline.

in the conversion analysis (Section 5). Per-cell EM and component ablations are reported in Appendix Tables 6 and 8.

Compass selector calibration moves closer to oracle while controlling latency. The deployment headline is a calibrated policy over the fixed family of evidence interfaces and decoding disciplines (Table 2). The claim is calibrated interface selection, not that PEC-HOP should replace Raw RAG in every reader–task cell. Compass selector improves macro EM over Raw RAG by $+9.36$, $+15.39$, and $+2.39$ percentage points on LFM2.5-1.2B, Qwen3-4B, and Gemma-4-E4B, respectively, at $2.02\times$, $5.69\times$, and $1.34\times$ latency speedups, and beats the single globally dev-selected policy on all three readers ($+1.97$, $+0.25$, $+5.99$ pp).

The per-query oracle is reported only as a non-deployable upper bound showing remaining headroom. On LFM2.5, Compass selector matches Dev-Argmax exactly (the lowest-burden admissible point happens to be the EM-max one); on Qwen3-4B Compass selector gives up 1.74 pp EM for a $1.4\times$ latency improvement over Dev-Argmax. Gemma-4-E4B shows a smaller but still positive gain, reinforcing that the safe choice is reader-dependent.

The selected policies are interpretable rather than uniformly compression-first: PEC-HOP is chosen where multi-passage bridge evidence is useful, Sentence-only where sentence-level evidence is sufficient, and Raw RAG when compression is not dev-safe.

The dev-budget sweep checks whether this deployment choice is tied to the particular first-50 calibration split. Across all budgets and all three readers, the calibrated policy stays above the single globally dev-selected policy (Figure 2); Table 2 separately shows the positive gains over Raw RAG for all three readers on the deterministic split, so the headline result is not a first-split artifact.

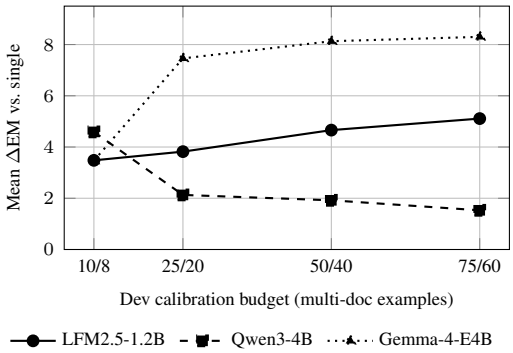

Figure 2: Dev-budget stability over 20 random partitions. Read the y-axis as the EM gain of Compass selector over one global dev-selected policy. The important pattern is not the exact slope but that all readers stay above zero at every budget; at the main $50/40$ budget, positive-partition counts are $20/20$, $17/20$, and $20/20$ for LFM2.5, Qwen3-4B, and Gemma.

Conversion links interface and decoding discipline. Since conversion depends on output shape as well as evidence content, the calibrated policy selects an interface–discipline pair. The full retrieval-base $\times$ decoding grids in Appendix G separate the two axes. On LFM2.5 at the same extractive discipline, PEC-HOP improves over the best non-PEC interface by $+1.0$ EM on HotpotQA, $+5.0$ on 2WikiMQA, and $+1.5$ on MuSiQue, while it is correctly not the preferred interface on the single-document tasks. Conversely, at a fixed Raw RAG interface, extractive decoding improves answer shaping but cannot choose the lowest-burden evidence representation. The frontier therefore comes from calibrating the evidence interface and decoding discipline together around conversion.

Reader	Policy	EM	F1	Lat. (ms)	Ctx tok	ΔEM vs Raw	Speed vs Raw
LFM2.5-Inst 1.2B	Raw RAG (default)	9.39	20.88	286.7	1356	–	1.00×
	Single dev-selected	16.78	26.46	178.3	834	+7.38	1.61×
	Dev-Argmax	**18.75**	**31.15**	**142.1**	**569**	+9.36	**2.02**×
	Compass selector	**18.75**	**31.15**	**142.1**	**569**	+9.36	**2.02**×
	Oracle	31.02	52.69	–	–	+21.62	–
Qwen3-4B	Raw RAG (default)	7.48	20.47	1410.6	1369	–	1.00×
	Single dev-selected	22.62	31.48	334.4	842	+15.14	4.22×
	Dev-Argmax	**24.61**	**38.67**	344.5	787	+17.13	4.09×
	Compass selector	22.87	35.75	**248.0**	**580**	+15.39	**5.69**×
	Oracle	37.03	58.69	–	–	+29.55	–
Gemma-4-E4B	Raw RAG (default)	19.19	29.98	385.3	1344	–	1.00×
	Single dev-selected	15.59	25.04	367.9	814	−3.60	1.05×
	Dev-Argmax	21.18	**34.43**	375.3	768	+1.99	1.03×
	Compass selector	**21.58**	34.40	**286.6**	**671**	+2.39	**1.34**×
	Oracle	32.81	52.03	–	–	+13.62	–

Table 2: Held-out macro results over five LongBench tasks for the fixed MCE candidate family. The Raw RAG row uses the default prompt as an uncompressed baseline; the Compass selector's ΔEM is reported against this default. Bold marks the best non-oracle value per reader and metric; Oracle is a non-deployable upper bound.

Policy class (LFM2.5-1.2B)	macro EM	Δ
Raw RAG + default (uniform)	9.04	—
Raw RAG + best fixed prompt (uniform ext)	13.76	+4.72
Prompt-only (Raw RAG, best prompt/task)	14.04	+5.00
Best single (base, prompt) uniform (PEC+ext)	15.76	+6.72
Interface-only (best base, fixed ext)	17.46	+8.42
Full-grid Argmax (oracle per cell)	17.84	+8.80
Compass selector (dev-calibrated, deterministic test split)	**18.75**	+9.71
Oracle per-query (within candidate pool)	31.02	+21.98

Table 3: Policy ablation; larger macro EM shows the value of interface selection beyond prompt choice. Oracle rows are on the full n=200/150 pool; the Compass selector row reports the dev-calibrated number on the held-out test split (the matched-split ΔEM of +9.36 over the same-split Raw RAG default is in Table 2). Bold marks the best deployable policy; the per-query oracle is shown only as an upper bound.

Interface and prompt are separable levers. Table 3 decomposes the LFM2.5 macro EM gain along the two calibration axes. Prompt selection alone (Raw RAG + best prompt per task, interface fixed) reaches 14.04 EM, a +5.00 pp gain over the default. Interface selection alone (best base per task at a fixed extractive prompt) reaches 17.46 EM, an *independent* +3.42 pp gain over the same prompt. Joint per-cell oracle selection adds a further +0.38 pp, reaching 17.84. The two axes therefore contribute almost additively: the evidence interface is a real lever that survives prompt control, and calibration over the interface is not a relabeling of prompt engineering. Demonstrations alone do not expose missing evidence; copy discipline helps only when the interface exposes the answer span.

Dataset	Evidence	Ans. rec.	Words	EM
HotpotQA	Raw article	66.06	280	17.76
HotpotQA	Extract. note	63.93	223	17.68
2WikiMQA	Raw article	54.64	292	–
2WikiMQA	Extract. note	51.34	196	–

Table 4: External-retrieval stress test. Focus on Ans. rec. versus Words: good evidence keeps answer recall high while shortening retrieved text. EM is reported only for the HotpotQA LFM2.5 reader run; 2WikiMQA is retrieval-only.

Extractive minimal evidence survives corpus-scale retrieval. To check that the gains are not only a bundled-LongBench artifact, we run a corpus-scale retrieval stress test over 66,581 HotpotQA articles. Table 4 shows the desired recall–budget tradeoff: extractive notes retain 96.8% of raw-article answer recall with 0.80× the words and statistically tied reader EM (17.68 vs 17.76). This partially validates the external-corpus setting for HotpotQA; the 2WikiMQA rows are retrieval-only and are reported only as supporting evidence for the same recall–budget direction.

5 Analysis

Design ablations. Dropping one design choice at a time from full PEC-HOP identifies which components carry the result (Appendix Table 10). The largest losses come from removing the card schema (−1.8 mean EM) or hydrated source window (−2.2), showing that the core mechanism is the combination of a structured card

schema with pointer-preserved hydrated source sentences. Anchor expansion is smaller on average (-0.3) but matters in the intended multi-hop cells (2WikiMQA -1.0, MuSiQue -1.5), where bridge terms determine whether the second hop is reachable. The hydration sweep gives the practical setting: w=0 is too narrow, while w=2 adds latency without quality gain. The corruption controls strengthen the same interpretation: PTR shuffle and random anchors both degrade PEC-HOP on multi-hop tasks (Appendix Table 17), so the cards are not merely sentence retrieval with extra formatting; source pointers and bridge anchors are load-bearing.

Conversion is a major marginal lever. Answer presence alone does not explain the EM gap; the key difference is whether answer-bearing predictions are shaped into exact answers. On LFM2.5, Raw RAG with its default prompt and the card primitive with the copy-oriented 2-shot extractive prompt have nearly identical loose-match rates (17.77% vs 17.63%), but their conversion $P(\text{EM} \mid \text{loose})$ differs sharply (43.41% vs 88.04%), yielding the observed $+6.7$ pp EM gap. Sentence-only retrieval with extractive decoding reaches an even higher conversion of 92.77% on the same reader (Appendix F); both copy-preserving interfaces lift conversion into the 88–93% range. The gain is therefore conversion-driven: Raw RAG often mentions the gold span, while the copy-oriented interface makes that span easier to emit in the scored form. This motivates conversion as a first-class SLM-RAG design lever alongside retrieval recall. The diagnostic is not formatting-only: the F1 guard protects long-form cells, and loose match separates answer visibility from exact answer shaping.

Conversion form is task-dependent. Short-answer QA benefits from extractive copying, while long-form QA needs a concise but complete phrase: a hard 5-word cap costs -5.97 F1 on multifieldqa_en. The Compass selector therefore keeps an F1 guard and selects a decoding discipline per cell, rather than treating conversion-aware decoding as "always shorter."

The best interface is cell-dependent. The selected cells are not explained by a single global compressor or decoding rule: LFM2.5 selects PEC-HOP on all three multi-hop tasks, while Qwen3-4B and Gemma-4-E4B keep Raw RAG on HotpotQA. Across readers, the two single-document tasks switch away from PEC-HOP toward Sentence-only. The safe low-burden interface therefore depends jointly on task structure, reader behavior, and answer form.

Sentence-only is the second mode, not just a baseline. On single-document QA, the answer usually appears in a local sentence rather than behind a second-hop entity, so Sentence-only keeps copy-preserving evidence while removing card overhead. It can even beat PEC-HOP (Qasper $+1.0$ EM, multifieldqa_en $+2.67$ EM on LFM2.5), reinforcing the need for a calibrated family rather than one fixed compressor.

6 Conclusion

Frozen SLM RAG is not limited by retrieval recall or context length alone. A retrieved passage can contain the answer and still fail if the frozen reader cannot convert that evidence into the exact answer span.

This paper treats the evidence interface as a two-stage design problem. PEC-HOP provides the copy-preserving compression stage: it is Pareto-best against LLMLingua-2 and Provence at 2.8–$4.8\times$ faster median latency, with no trained model on the critical path. MCE-Compass then calibrates the (interface, prompt) operating point per (reader, task) cell rather than treating PEC-HOP as a universal Raw RAG replacement, improving macro EM over Raw RAG by $+9.36$, $+15.39$, and $+2.39$ pp on LFM2.5, Qwen3-4B, and Gemma-4-E4B, and beating the single globally dev-selected policy. The broader lesson is that for frozen SLM readers, evidence must remain faithful enough to copy, formatted enough to convert, and calibrated enough to avoid unsafe compression.

Future work should move from cell-level calibration to instance-level routing, so the system can decide per question when to use PEC-HOP, Sentence-only evidence, or a Raw RAG fallback while preserving explicit accuracy guarantees.

Limitations

Statistical resolution and dev-split stability. Each LongBench task has ≤ 200 test items, which limits paired-bootstrap resolution to roughly ± 4 pp on EM for single task cells. The strongest claims should therefore be read as macro-aggregate claims over three multi-hop tasks

(compressor head-to-head) and five tasks (Compass selector headline). For reproducibility the main tables use deterministic first-50 dev splits; the 20-partition dev-budget sweep in Figure 2 checks that the calibrated policy stays above the single globally dev-selected policy across budgets and readers, but larger dev sets would give tighter cell-level decisions.

Reader and benchmark scope. The protocol is evaluated on three frozen reader families (LFM2.5, Qwen3, and Gemma) and English LongBench-style QA tasks. Broader model families, larger readers, instruction-tuning variants, and non-English settings remain untested. The results should therefore be read as evidence for a deployment-time calibration protocol, not as a universal zero-shot routing rule.

Calibration requirements. Compass selector needs a small labeled calibration set (~ 50 examples per (reader, task) cell) to estimate dev EM, F1, latency, and conversion. This is realistic for enterprise RAG settings with logged Q&A pairs or domain-expert annotation, but it is unavailable in fully zero-shot deployments. The framework also assumes that a task tag or task cell is known at inference time. The rule uses fixed coefficients and guards rather than learned hyperparameters; Appendix C reports a sensitivity check, but larger calibration sets would support tighter selection. Moving from cell-level calibration to per-instance routing is left for future work.

Retrieval and evidence-construction scope. All main experiments use a single frozen retrieval encoder (gte-small), and the card representation relies on lightweight lexical anchors. This keeps the method non-parametric and cheap, but it can miss lower-cased relations or domain-specific bridge cues; trained card encoders, richer relation extraction, and cross-lingual retrieval are not explored. The corpus-scale stress test (Table 4, Appendix H) covers one external dataset and one reader, so a full external-corpus version of the three-reader Compass selector grid remains future work.

Runtime and implementation scope. Latency is measured on an RTX 3090 workstation with quantized GGUF readers under `llama.cpp`. Absolute latencies will vary across hardware, runtimes, and quantization choices; the claim is the relative effect of evidence representation under a fixed implementation. In LongBench the relevant passages are bundled, so card construction and encoding are included in request-time latency. In a fixed external-corpus deployment the card index would be built once, but that regime is not fully benchmarked here.

Metric and decoding dependence. The conversion diagnostic is tied to EM/loose-match behavior and is most directly suited to extractive or short-answer QA. The goal is not to optimize EM formatting alone: verbose-fail cases are counted through loose match, and the F1 guard prevents short answers from winning when long-form content is required. Free-form generation, multi-sentence outputs, or non-extractive evaluation may require different conversion diagnostics, prompt families, and selection guards.

References

Alexander Amini, Anna Banaszak, Harold Benoit, Arthur Böök, Tarek Dakhran, Song Duong, Alfred Eng, Fernando Fernandes, Marc Härkönen, Anne Harrington, et al. 2025. Lfm2 technical report. *arXiv preprint arXiv:2511.23404*.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, et al. 2024. Longbench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the 62nd annual meeting of the association for computational linguistics (volume 1: Long papers)*, pages 3119–3137.

Xin Cheng, Xun Wang, Xingxing Zhang, Tao Ge, Si-Qing Chen, Furu Wei, Huishuai Zhang, and Dongyan Zhao. 2024. xrag: Extreme context compression for retrieval-augmented generation with one token. *Advances in Neural Information Processing Systems*, 37:109487–109516.

Nadezhda Chirkova, Thibault Formal, Vassilina Nikoulina, and Stéphane Clinchant. 2025. Provence: efficient and robust context pruning for retrieval-augmented generation. *arXiv preprint arXiv:2501.16214*.

Yixiong Fang, Tianran Sun, Yuling Shi, and Xiaodong Gu. 2025. Attentionrag: Attention-guided context pruning in retrieval-augmented generation. *arXiv preprint arXiv:2503.10720*.

Gemma Team and Google DeepMind. 2026. Gemma 4 model card. https://ai.google.dev/gemma/docs/core/model_card_4.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th Inter-*

national Conference on Computational Linguistics, pages 6609–6625.

Chris Hokamp and Qun Liu. 2017. Lexically constrained decoding for sequence generation using grid beam search. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1535–1546.

Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 874–880.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2023. Atlas: Few-shot learning with retrieval augmented language models. *Journal of Machine Learning Research*, 24(251):1–43.

Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong C Park. 2024. Adaptive-rag: Learning to adapt retrieval-augmented large language models through question complexity. 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 7036–7050.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2023. Llmlingua: Compressing prompts for accelerated inference of large language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pages 13358–13376.

Huiqiang Jiang, Qianhui Wu, Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2024. Longllmlingua: Accelerating and enhancing llms in long context scenarios via prompt compression. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1658–1677.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. 2022. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*.

Amita Kamath, Robin Jia, and Percy Liang. 2020. Selective question answering under domain shift. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 5684–5696.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pages 6769–6781.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

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

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.

Maxime Louis, Hervé Déjean, and Stéphane Clinchant. 2025. Pisco: Pretty simple compression for retrieval-augmented generation. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 15506–15521.

Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Menglin Xia, Xufang Luo, Jue Zhang, Qingwei Lin, Victor Rühle, Yuqing Yang, Chin-Yew Lin, et al. 2024. Llmlingua-2: Data distillation for efficient and faithful task-agnostic prompt compression. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 963–981.

Matt Post and David Vilar. 2018. Fast lexically constrained decoding with dynamic beam allocation for neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1314–1324.

Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher Manning. 2024. Raptor: Recursive abstractive processing for tree-organized retrieval. In *International Conference on Learning Representations*, volume 2024, pages 32628–32649.

Wenhan Xiong, Xiang Lorraine Li, Srini Iyer, Jingfei Du, Patrick Lewis, William Yang Wang, Yashar Mehdad, Wen-tau Yih, Sebastian Riedel, Douwe Kiela, et al. 2020. Answering complex open-domain questions with multi-hop dense retrieval. *arXiv preprint arXiv:2009.12756*.

Fangyuan Xu, Weijia Shi, and Eunsol Choi. 2023. Recomp: Improving retrieval-augmented lms with compression and selective augmentation. *arXiv preprint arXiv:2310.04408*.

Shi-Qi Yan, Jia-Chen Gu, Yun Zhu, and Zhen-Hua Ling. 2024. Corrective retrieval augmented generation. *arXiv preprint arXiv:2401.15884*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao,

Chengen Huang, Chenxu Lv, et al. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 2369–2380.

Wenhao Yu, Hongming Zhang, Xiaoman Pan, Peixin Cao, Kaixin Ma, Jian Li, Hongwei Wang, and Dong Yu. 2024. Chain-of-note: Enhancing robustness in retrieval-augmented language models. In *Proceedings of the 2024 conference on empirical methods in natural language processing*, pages 14672–14685.

A Related-Work Design Axes

Method	Stage	Trained component	Reader change	Verbatim spans	Multi-hop capable
LLMLingua-2 (Pan et al., 2024)	query-time	570M XLM-R	none	tokens	no
Provence (Chirkova et al., 2025)	query-time	430M DeBERTa CE	none	sentences	no
AttentionRAG (Fang et al., 2025)	query-time	none (attn pass)	extra fwd	partial	no
PISCO (Louis et al., 2025)	offline	distilled comp.	none	no	no
xRAG (Cheng et al., 2024)	offline	bridge + reader FT	yes	no	no
MCE family	offline	*none*	none	sentences + window	✓

Table 5: Design-axis comparison against the closest compression baselines; the key entry is the no-training, no-reader-change, verbatim-span, multi-hop-capable row.

The key design distinction is whether a method preserves reader-visible verbatim spans without adding a trained component or changing the reader. Table 5 summarizes these axes; the MCE family of copy-preserving interfaces is the only row with all three properties, which separates it from query-time compressors and learned bridge methods.

	LFM2.5-1.2B			Gemma-4-E4B			Qwen3-4B		
Method	Hot	2Wi	MuS	Hot	2Wi	MuS	Hot	2Wi	MuS
Raw RAG	24.0	25.0	8.0	14.5	8.0	8.5	20.5	25.5	4.0
Sentence-only	22.0	23.0	7.5	5.5	1.0	2.5	19.0	28.0	5.5
LLMLingua-2 (Pan et al., 2024)	23.5	25.5	5.5	8.0	8.5	6.0	18.5	26.5	9.5
Provence (Chirkova et al., 2025)	28.5	25.5	11.5	14.0	5.0	7.5	27.5	31.5	8.0
PEC-HOP (full)	**27.5**	**32.5**	**9.5**	**24.5**	**30.5**	**15.0**	**32.0**	**39.5**	**20.0**
– anchor-overlap rerank	27.0	30.5	10.0	26.5	28.0	13.5	33.0	33.0	18.5
– 2-shot prompt	26.5	31.0	11.5	11.5	7.5	8.5	21.0	29.5	7.0
– both	25.5	30.0	9.5	14.0	8.0	7.5	20.5	23.0	4.5

Table 6: Per-cell EM backing Table 1: three multi-hop LongBench tasks × three readers, extractive decoding throughout. PEC-HOP (full) wins outright against the strongest external compressor (Provence) on 7 of 9 cells (the LFM2.5 HotpotQA and MuSiQue cells are within 1−2 pp). On 2WikiMQA, the gain over Provence is largest at +7.0 pp (LFM), +25.5 pp (Gemma), and +8.0 pp (Qwen3): explicit bridge expansion via card anchors is the lever Provence's single-pass cross-encoder misses. Rows below the bold full method ablate one of PEC-HOP's two non-parametric components; the per-component aggregate is in Table 8.

	LFM2.5-1.2B			Gemma-4-E4B			Qwen3-4B		
Method	Hot	2Wi	MuS	Hot	2Wi	MuS	Hot	2Wi	MuS
LLMLingua-2 (Pan et al., 2024)	1019	1019	1019	1281	1281	1281	867	867	867
Provence (Chirkova et al., 2025)	654	654	654	644	644	644	540	540	540
PEC-HOP	**126**	**126**	**161**	**268**	**247**	**286**	**251**	**267**	**264**

Table 7: Median end-to-end latency (ms) per reader–task cell (n=200 queries each) accompanying Table 1. The trained-compressor rows repeat the per-reader medians from Table 1; the macro medians used for speedups average over the nine reader–task cells. PEC-HOP is the fastest method in 9 of 9 cells, with macro median 222 ms vs 613 for Provence (2.8× faster) and 1056 for LLMLingua-2 (4.8×). The speedup compresses on Qwen3-4B because reader prefill dominates the cost there, while compressor models still contribute a ∼ 300 ms fixed overhead. PEC-HOP's only critical-path addition over Raw RAG is one extra gte-small encoder pass for the hop-2 query (∼5−10 ms).

Variant	LFM	Gemma	Qwen3	Macro
PEC-HOP (full)	**23.17**	**23.33**	**30.50**	**25.67**
– anchor-overlap rerank	22.50	22.67	28.17	24.44
– 2-shot prompt	23.00	9.17	19.17	17.11
– both (cards-only + extractive)	21.67	9.83	16.00	15.83

Table 8: Ablation of PEC-HOP's two non-parametric components. Each row removes one (or both) of the components from the full method. The 2-shot prompt is the dominant component on Gemma-4-E4B (−14.16 pp when removed) and Qwen3-4B (−11.33 pp), which produce verbose answers without demonstrations; both readers' loose-match rate is high but conversion to Exact Match fails. On LFM2.5, which already cooperates with short-span instructions, the anchor-overlap rerank is the larger of the two small effects (−0.67 pp for rerank, −0.17 pp for the 2-shot prompt). The combined drop is −1.5 pp on LFM, −13.5 pp on Gemma, and −14.5 pp on Qwen3, indicating mild interaction rather than strict additivity between the two components.

The full PEC-HOP method combines non-parametric evidence construction with a fixed conversion-aware decoding discipline. Table 8 removes these components one at a time. Stronger copy discipline is the dominant lever on the two verbose-tendency readers (Gemma, Qwen3), where conversion to Exact Match—not retrieval—is the bottleneck even when correct evidence is present. On LFM2.5, which already follows short-span instructions, the evidence-construction and decoding components contribute more evenly.

Method	LFM2.5-1.2B		Gemma-4-E4B		Qwen3-4B	
	0-shot	2-shot	0-shot	2-shot	0-shot	2-shot
Raw RAG	19.00	20.17	10.33	**24.00**	16.67	**34.17**
Sentence-only	17.50	17.67	3.00	15.33	17.50	22.33
LLMLingua-2 (Pan et al., 2024)	18.17	18.83	7.50	18.67	18.17	26.67
Provence (Chirkova et al., 2025)	21.83	21.17	8.83	21.67	**22.33**	29.33
PEC-HOP (ours)	**23.00**	**23.17**	9.17	23.33	19.17	30.50

Table 9: Compressor ranking under matched extractive prompts (0-shot vs 2-shot, identical template aside from the two demonstrations). Macro EM across HotpotQA, 2WikiMQA, and MuSiQue (n=200 each). Bold marks the best method per (reader, prompt) cell. On LFM2.5-1.2B, PEC-HOP is the best method under *both* prompts, so its head-to-head advantage in Table 1 is not prompt-conditional. On Gemma-4-E4B and Qwen3-4B the ranking shifts substantially across prompts and Raw RAG becomes the strongest cell under 2-shot: the demonstrations add up to +13.67 pp on Gemma Raw RAG and +17.50 pp on Qwen3 Raw RAG, while adding only +0.17 pp on LFM2.5 PEC-HOP. This confirms the two-bottleneck reading of Section 5: conversion is the dominant lever on Gemma and Qwen3, and the evidence interface is the dominant lever on LFM2.5.

Table 9 extends Table 8 to all compressors under the same 0-shot vs 2-shot extractive prompts. On LFM2.5, PEC-HOP stays the best method under both prompts (23.00 at 0-shot, 23.17 at 2-shot), so its head-to-head advantage in Table 1 reflects the evidence interface itself rather than an interaction with the prescribed 2-shot template. On Gemma, the 2-shot demonstrations add roughly +11 to +14 pp across methods. On Qwen3, gains are more method-dependent, ranging from about +5 to +18 pp, with the largest gains on Raw RAG and PEC-HOP. LFM moves by less than +1 pp on every method, mirroring the reader-dependent conversion gap described in Section 5. The two tables together separate the compressor's marginal effect from the prompt's marginal effect on each reader.

B Single-Axis Ablations

Variant	EM	F1	Lat (ms)	Ctx tok	gold-in-ctx (%)
Card-primitive component ablations					
PEC cards (full, no rerank/2-shot)	10.5	22.4	140	838	37.4
– anchor expansion	10.2	21.8	146	844	35.2
– card schema	8.7	20.8	101	349	27.5
– hydration	8.3	20.7	132	686	31.0
Hydration window sweep					
w=0 (sentence)	8.5	20.1	108	410	–
w=2 (wider)	9.8	21.9	245	1187	–

Table 10: Single-axis ablations of the card primitive (no rerank, no 2-shot prompt) on LFM2.5-1.2B; smaller drops indicate less essential components. The full PEC-HOP method adds anchor-overlap rerank and a 2-shot extractive prompt on top of this primitive (see Table 8).

The single-axis ablation identifies which parts of the card primitive are load-bearing. Relative to the full primitive (10.5 EM, 22.4 F1, 140 ms, 838 context tokens), removing anchor expansion is mild (10.2 EM), but removing the card schema drops to 8.7 EM and removing hydration drops to 8.3 EM. A wider w=2 hydration window raises latency and context without improving EM. The full PEC-HOP method further adds anchor-overlap rerank and a 2-shot extractive prompt; those additions are ablated in Table 8.

C Compass Sensitivity

The Compass selector operating point uses three fixed guards: the 0.35 token-burden coefficient in $B(c)$, the 0.10 log-burden tie band, and the 3 pp F1 guard. These values are not tuned on test data. We varied the token coefficient in a small neighborhood around 0.35, widened and narrowed the burden band around 0.10, and checked adjacent F1 guards. The selected cells change only at near-ties on the dev frontier, and the macro pattern remains the same: the calibrated policy keeps a positive Raw RAG gain while preferring lower-burden candidates when accuracy is non-inferior. The F1 guard is the most important safety valve for non-short-answer tasks: removing it lets overly short extractive answers win some long-form cells by EM or latency, whereas the reported rule keeps those candidates out unless F1 remains non-inferior.

D Reproducibility

Models and quantization. All readers are loaded as Q4_K_M GGUF quantized weights through `llama.cpp` (commit pinned in the code release): LFM2.5-Instruct 1.2B (`LiquidAI/LFM2.5-1.2B-Instruct`, hybrid convolutional neural network/grouped-query attention), Qwen3-4B (`Qwen/Qwen3-4B`), and Gemma-4-E4B-it (`google/gemma-4-E4B-it`). Quantized GGUF artifacts are recorded with source repos, filenames, and SHA-256 hashes in the code-release manifest. Reader runtime: `n_ctx=2048`, all layers offloaded to a single NVIDIA RTX 3090 (24 GB memory), CUDA 12.1, no draft model, no speculative decoding.

Decoding parameters. Greedy decoding (temperature 0, top-p 1, top-k 0, no repetition penalty) for all variants and all readers, to remove sampling variance from the comparison. Maximum new tokens: 80 for short-answer prompts (Default, Extractive, Short15) and 128 for Concise. Stop se-

quences: `\n\n`, `Answer:`, `Question:`. The full prompt strings are in Appendix E.

Retrieval encoder and indices. A single frozen sentence encoder, gte-small, is used (`thenlper/gte-small`, 384-dim), with normalized inner product similarity over a per-question FAISS `IndexFlatIP`. Sentences are split with the Natural Language Toolkit `punkt` tokenizer; cards are emitted per the rules in Section 3.1 (K_a=10 anchors per card, K_c=12 cards per passage, sentences with $<$ 5 word tokens or no extractive signal skipped).

Anchor regex. ANCHORS are extracted by Python regex against t and s_j in priority order:

1. multi-word capitalized phrases: `[A-Z][a-z]+(?:\s+[A-Z][a-z]+)+`
2. single capitalized tokens not at sentence start: `(?<=\s)[A-Z][a-z]+`
3. four-digit years: `\b(1[5-9][0-9]{2}|20[0-9]{2})\b`
4. numeric thresholds: `\b\d+(?:\.\d+)?(?:\%|\,000+)?\b`

The limitation that lower-cased relations (e.g., "directed by") are not captured is discussed in Limitations.

Dataset splits. The LongBench (Bai et al., 2024) English subsets are used as shipped, taking the first 200 instances per task (150 for multifieldqa_en, the size of that task). For the Compass selector selection (Section 4.1), the first 50 instances of each task (40 for multifieldqa_en) form the dev split and the remainder forms the held-out test split. The split is deterministic (no shuffling), so re-runs with the same code reproduce identical dev/test partitions. Figure 2 quantifies dev-budget sensitivity over 20 random partitions at four budgets.

Latency measurement. End-to-end latency is the wall-clock time from submitting the question to emission of the answer span, measured with `time.perf_counter` around a single Python call that includes (i) sentence-encoder forward pass for q (and q' for PEC-HOP), (ii) FAISS retrieval, (iii) reader prefill, and (iv) reader decode. A 5-question warmup is run before timing to flush just-in-time compilation and cache effects. Unless a table explicitly states median latency, latency is the per-cell mean over test instances; macro latency then averages task cells equally, and speedup is the Raw RAG macro latency divided by the comparison row's macro latency. Table 1 and Table 7 instead use median latency for the matched-discipline compressor head-to-head. Qwen3-4B Raw RAG has heavy-tailed prefill outliers, which is why median latency is preferred for the compressor head-to-head.

Significance testing. Paired bootstrap with 10^4 resamples, paired by question index. The non-inferiority margin is pre-specified at -3 pp on both EM and F1 for any non-inferiority claim.

Code and artifacts. Evaluation scripts, prompt strings, anchor regex, FAISS index builder, and result-aggregation code will be released upon acceptance. Random seeds are set to 0 for any sampling-related operation (none on the critical path under greedy decoding).

Compute. Total compute for the experiments reported: $\sim$ 200 GPU-hours on a single RTX 3090 GPU. The largest measured cost is Qwen3-4B reader inference on Raw RAG variants (8+ seconds median per question on multi-hop tasks).

Licenses. LongBench is released under MIT; LFM2.5-1.2B-Instruct is released under the LFM Open License Agreement; Qwen3-4B and Gemma-4-E4B-it are released under Apache 2.0. gte-small is released under MIT. The use of the listed resources is consistent with the applicable licenses.

E Decoding Prompt Templates

Default. "Answer the following question based on the provided documents. Give a short, direct answer."
Extractive (5-word cap). "Read the short evidence snippets and answer the question by copying the shortest span from the evidence that directly answers it. Reply with at most 5 words. Do not write a full sentence."
Short15 (15-word cap). Identical to Extractive but with a 15-word cap.
Concise (no cap). "Answer the following question based on the provided documents. Be concise: prefer the exact answer span over a full sentence." Label is the default "Answer:".
2-shot extractive (used by PEC-HOP). Identical to Extractive, prefixed with two fixed task-

Base	Prompt	loose	conversion
Raw RAG	default	17.77%	43.41%
Raw RAG	concise	16.53%	53.55%
Sentence-only	extractive	16.37%	**92.77%**
PEC cards	extractive	17.63%	**88.04%**

Table 11: Conversion decomposition on LFM2.5-1.2B, macro across five LongBench tasks. *loose* is the rate at which the prediction contains an acceptable answer string after normalization; *conversion* = $P(\mathrm{EM}{=}1 \mid \mathrm{loose}{=}1)$ is the fraction of answer-bearing predictions that are shaped as exact matches. Raw RAG and the two copy-preserving interfaces produce answer-bearing predictions at nearly the same loose-match rate, but their conversion differs by 35–49 percentage points; both copy-preserving interfaces fall in the 88–93% conversion range on this reader.

agnostic demonstrations of bridge-style copy answers (an Einstein–Germany example and an *Inception*-director birth-date example). The two demonstrations add ∼140 reader tokens and are reused verbatim across all tasks and readers. Reducing to a single demonstration triggered multi-minute decodes on a small fraction of Qwen3-4B queries; two demonstrations are therefore the prescribed default. The full demonstration strings are released with the accompanying code.

F Conversion Diagnostic

Table 11 decomposes the LFM2.5 macro EM into loose-match and conversion. Raw RAG with the default prompt reaches only 43.41% conversion; both copy-preserving interfaces (sentence-only and PEC cards) under an extractive decoding discipline land in the 88–93% range while keeping similar loose-match rates. This supports the conversion-driven framing in Section 5.

G Cross-Prompt Grids over 0-shot Decoding Disciplines

Base	Prompt	HotpotQA				2WikiMQA				MuSiQue				Qasper				MFQA-en			
		def	ext	s15	con	def	ext	s15	con	def	ext	s15	con	def	ext	s15	con	def	ext	s15	con
EM																					
Raw RAG		18.0	24.0	22.0	20.0	18.0	25.0	24.5	20.5	4.5	8.0	8.0	6.5	0.0	8.5	8.0	1.0	4.7	3.3	2.0	2.7
Sentence-only		18.0	22.0	22.0	19.0	17.0	23.0	22.5	18.0	6.0	7.5	6.0	6.5	3.0	**13.0**	9.0	4.0	8.7	9.3	**10.7**	8.0
PEC cards		20.5	25.5	**26.0**	22.5	17.0	**30.0**	29.5	21.0	7.0	**9.5**	**9.5**	7.5	2.0	8.5	6.0	1.5	6.0	5.3	6.0	6.7

Table 12: LFM2.5 cross-prompt grid over Compass candidate bases (Raw RAG, Sentence-only, PEC cards) × four decoding prompts. The card row is the primitive interface that underlies PEC-HOP; the full PEC-HOP adds rerank and 2-shot extractive prompt (Table 1). Bold cells identify the best interface–prompt choice per task.

Base	Prompt	HotpotQA				2WikiMQA				MuSiQue				Qasper				MFQA-en			
		def	ext	s15	con	def	ext	s15	con	def	ext	s15	con	def	ext	s15	con	def	ext	s15	con
EM																					
Raw RAG		15.0	20.5	12.5	29.0	8.5	25.5	8.0	16.5	7.5	4.0	1.0	**13.0**	2.5	6.0	3.0	5.5	4.0	4.0	3.3	4.7
Sentence-only		13.5	19.0	13.0	14.5	10.5	**28.0**	9.0	14.0	2.5	5.5	4.0	5.0	5.0	**10.5**	4.0	4.0	3.3	7.3	7.3	8.0
PEC cards		19.0	20.5	12.0	**31.0**	6.5	23.0	8.5	22.5	7.5	4.5	3.5	11.5	3.5	6.0	4.0	6.5	4.0	5.3	6.0	**9.3**

Table 13: Qwen3-4B cross-prompt grid; the card row is the primitive interface that underlies PEC-HOP. Bold cells identify the best interface–prompt choice per task.

The cross-prompt grids report the four 0-shot decoding disciplines (default, extractive, short15, concise); the fifth Compass candidate—the 2-shot extractive prompt—is the prescribed PEC-HOP configuration and is reported in Tables 1 and 8 instead. The grids show why a single global interface–prompt choice is unsafe. The bold cells in Tables 12 and 13 move across tasks, readers, interfaces, and prompts. On LFM2.5, the card primitive is strongest on multi-hop tasks under extractive or Short15 decoding, while Sentence-only is often better on single-document tasks. Qwen3-4B shifts some good card cells toward concise decoding. These grids show the cards-only primitive; the full PEC-HOP method (Tables 1, 2) further adds anchor-overlap rerank and a 2-shot extractive prompt.

H Full-Retrieval Evidence Recall

Variant	Ans. recall	Supp. title	All titles	Supp. sent.	Words	Ans. dens./1k
Raw article	66.06	88.42	80.95	74.34	280	2.587
Raw chunk-100	65.06	88.19	80.43	72.09	240	2.843
Raw chunk-40	58.68	86.47	76.93	53.40	137	4.332
Extractive minimal note	63.93	87.43	79.08	70.60	223	2.981
Smart minimal note	61.73	87.04	78.35	61.11	228	2.829
LLM entity note	46.14	82.78	70.93	0.45	263	1.765

Table 14: HotpotQA retrieval-only recall; good evidence preserves answer recall with fewer words.

The full-retrieval recall check asks whether extractive evidence keeps answer coverage while reducing the evidence budget. In Table 14, raw articles have 66.06 answer recall with 280 words; extractive minimal notes keep 63.93 answer recall, or 96.8% of raw recall, while using 223 words (0.80×). The LLM entity-note row is the negative control: it falls to 46.14 answer recall and nearly loses supporting-sentence coverage.

Evidence index	n	EM	F1	ΔEM vs Raw	95% CI	ΔF1 vs Raw
Raw article	7404	17.76	29.68	–	–	–
Extractive minimal note	7405	17.68	29.57	−0.08	[−0.62, +0.47]	−0.11
Smart minimal note	7405	17.57	29.28	−0.19	[−0.81, +0.43]	−0.39
LLM entity note	7405	13.32	24.30	−4.44	[−5.25, −3.63]	−5.37

Table 15: HotpotQA external-retrieval reader EM; good compression stays statistically tied with raw articles.

The external-reader EM check tests whether the shorter extractive evidence is performance-neutral

after retrieval. In Table 15, extractive minimal notes are statistically tied with coverage-matched raw article retrieval: EM is 17.68 versus 17.76, with ΔEM $= -0.08$ and CI $[-0.62, +0.47]$. The LLM entity-note row is the negative control, dropping to 13.32 EM and ΔEM $= -4.44$.

Variant	Ans. recall	Supp. title	All titles	Supp. sent.	Words	Ans. dens./1k
Raw article	54.64	94.47	87.25	65.38	292	2.330
Extractive minimal note	51.34	93.14	84.37	59.74	196	2.806
LLM entity note	46.60	88.07	72.67	1.12	263	1.790

Table 16: 2WikiMQA retrieval-only recall; good evidence preserves answer recall with fewer words.

The 2WikiMQA transfer check follows the same recall–budget pattern as HotpotQA. In Table 16, raw articles have 54.64 answer recall with 292 words, whereas extractive minimal notes keep 51.34 answer recall (94.0% of raw recall) with only 196 words (0.67×). Reader EM is not run for 2WikiMQA, so this is a retrieval-only transfer check.

I Corruption Stress Test

Variant	HotpotQA		2WikiMQA		MuSiQue	
	EM	F1	EM	F1	EM	F1
PEC cards (primitive)	**20.5**	**31.1**	17.0	**26.5**	**7.0**	**14.3**
PTR shuffle	14.0	22.7	**18.0**	26.4	6.5	12.1
Random anchors	18.0	26.6	15.5	24.1	3.5	10.1

Table 17: Corruption stress test; larger drops identify load-bearing card fields.

The corruption stress test checks whether card fields are load-bearing rather than decorative. In Table 17, PTR shuffle tests whether source-sentence pointers matter, while random anchors test whether bridge terms matter. On HotpotQA, PTR shuffle drops EM from 20.5 to 14.0, making the source pointer the strongest lever. On MuSiQue, random anchors drop EM from 7.0 to 3.5, so bridge expansion is more important. The asymmetric degradation argues that the card primitive is not just plain sentence retrieval.