RETRIEVAL-AWARE VOCABULARY FOLDING FOR QUERY-SIDE COMPRESSION IN ASYMMETRIC LLM RETRIEVAL

Anonymous authors
Paper under double-blind review

ABSTRACT

Asymmetric LLM retrievers reduce query latency by replacing online query-side LLM inference with a lightweight EmbeddingBag lookup. However, they retain a full tokenizer-aligned query vocabulary, introducing a large memory footprint despite the sparse usage of real retrieval queries. We reveal that this redundancy is not equivalent to token redundancy: rare tokens may carry critical lexical evidence for ranking, making naive vocabulary pruning unsafe. We propose **Retrieval-Aware Vocabulary Folding (RAVF)**, a retrieval-aware vocabulary folding framework that compresses the query-side EmbeddingBag through utility-based anchor selection, token protection, and evidence-preserving folding. Instead of deleting removed tokens, **RAVF** maps them to compatible anchors while keeping the document encoder and retrieval index unchanged. Experiments on BEIR, CMTEB-R, and multiple LLM backbone scales show that **RAVF** significantly improves the accuracy–memory trade-off of asymmetric retrieval. Ablation studies further demonstrate that retrieval-aware folding and token protection are the main contributors, while lightweight InfoNCE calibration is sufficient without specialized distillation objectives.

1 INTRODUCTION

LLM retrievers achieve strong ranking quality but require costly online query encoding (Wang et al., 2024; BehnamGhader et al., 2024; Lee et al., 2025). Asymmetric systems move LLM inference to offline document encoding and use an EmbeddingBag for dense queries; a sparse branch can retain exact lexical matches (Ma et al., 2026). The dense EmbeddingBag still stores a row for every tokenizer entry, although typical queries use only a small fraction of them.

This split retains the semantic capacity of LLM-derived document vectors while making online dense query encoding a lookup operation. The sparse branch separately preserves exact matches for entities and other lexical constraints. RAVF targets the remaining dense query vocabulary, so it can be applied without changing either the cached document representations or the sparse matching path.

Removing infrequent entries is unsafe: rare entities, numbers, and specialist terms can determine which document is relevant. Our preliminary analysis confirms both sparse vocabulary usage and uneven retrieval importance. Query-side compression must therefore reduce stored rows while preserving evidence carried by removed tokens.

Unlike compressing the LLM encoder or cached document vectors, this target sits directly in the online query path. A smaller EmbeddingBag reduces the dense branch's stored parameters without rebuilding the document collection. However, deleting a tokenizer row also deletes that token's contribution to every future query, even when the token is rare only because it names a specific entity or constraint.

We propose **Retrieval-Aware Vocabulary Folding (RAVF)** (Figure 1). RAVF selects anchors using retrieval utility, protects critical lexical tokens, and maps other tokens to compatible anchors instead of deleting them. Optional InfoNCE calibration (van den Oord et al., 2018) updates only the compact query EmbeddingBag; the tokenizer, document encoder, descriptors, and index remain fixed.

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.

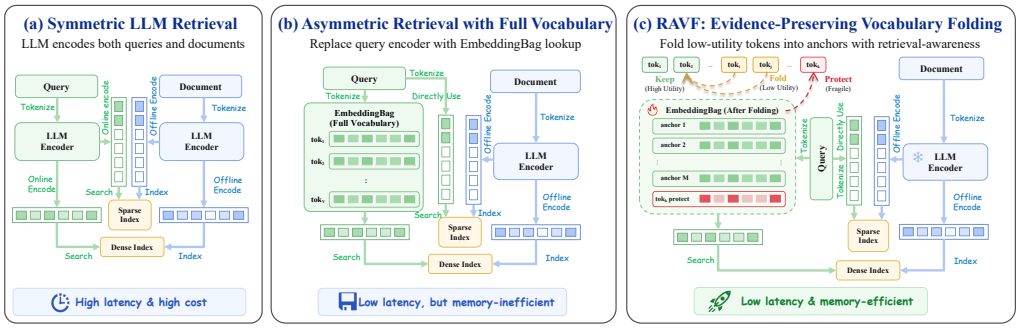


Figure 1: Motivation and overview of RAVF. (a) Symmetric LLM retrieval requires expensive online query encoding. (b) Asymmetric dense retrieval removes query-side LLM inference but retains a large tokenizer-aligned EmbeddingBag. (c) RAVF compresses the query vocabulary through retrieval-aware anchor selection and evidence-preserving token folding, reducing query-side redundancy while maintaining the original document space.

Across BEIR, CMTEB-R, and multiple LLM scales, RAVF improves the accuracy–memory trade-off and remains compatible with hybrid retrieval. The analysis and ablations establish the query vocabulary as a compression target and show why retrieval-aware selection and evidence-preserving folding matter.

Our contributions are:

- We identify the query vocabulary interface as an overlooked compression bottleneck in asymmetric LLM retrieval. And reveal that query vocabulary usage is highly sparse while token retrieval importance is highly non-uniform: frequency-based pruning may remove tokens that carry critical ranking evidence.
- We propose RAVF, a retrieval-aware vocabulary folding framework that compresses the query EmbeddingBag while preserving removed token information through anchor folding.
- Extensive experiments across multiple benchmarks and model scales demonstrate that RAVF improves the accuracy–memory trade-off of asymmetric LLM retrieval, while remaining compatible with hybrid retrieval systems that combine dense semantic matching and sparse lexical matching.

2 Related Work

LLM and asymmetric retrieval. Dense retrievers rank document vectors against query vectors (Karpukhin et al., 2020). LLM encoders improve those vectors but raise online latency (Wang et al., 2024; Lee et al., 2025). LightRetriever (Ma et al., 2026) instead encodes documents offline and uses a query EmbeddingBag. Its remaining dense query cost is dominated by tokenizer-aligned embedding rows. RAVF compresses this interface while retaining the document index and the optional sparse branch.

Vocabulary compression. Random, frequency, and IDF selection (Jones, 1972) and learned magnitude or sensitivity criteria (Han et al., 2016; Molchanov et al., 2017) discard selected rows. Rare terms can nevertheless be decisive for retrieval. Late-interaction compression reduces contextual token representations after encoding (Khattab & Zaharia, 2020; Santhanam et al., 2022); RAVF instead folds tokens before query descriptor construction so their contributions remain available.

These approaches act at different stages. A low-frequency rule favors common tokens, whereas IDF favors specificity; neither directly measures whether a token changes relevant-document ranking. Magnitude and Taylor criteria assess parameters or loss sensitivity but still remove the chosen token rows. Folding preserves a path from every original token ID to a retained row, which is important when the tokenizer itself must stay compatible with an existing retrieval pipeline.

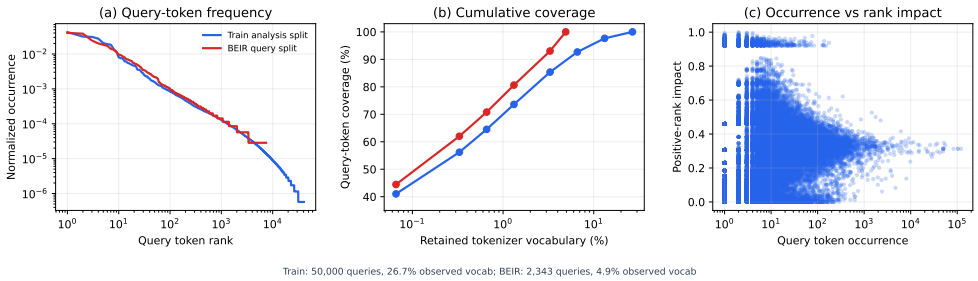


Figure 2: Preliminary analysis of query-side vocabulary compression. (a) Query vocabulary follows a long tail. (b) Most queries use a small vocabulary subset. (c) Rare tokens can affect retrieval ranking.

Retrieval calibration. InfoNCE optimizes positive scores against negatives (van den Oord et al., 2018; Karpukhin et al., 2020). With fixed document descriptors, it directly calibrates the compact EmbeddingBag for the inference score $s(q, d) = q^\top d$. RAVF relies on vocabulary transformation rather than a specialized distillation loss.

3 Preliminary Analysis

Before introducing RAVF, we test whether the query EmbeddingBag has unused rows and whether frequency predicts retrieval value. The analysis uses a LightRetriever-style Qwen2.5-1.5B model (Yang et al., 2024) with 151,666 query token rows; document representations are encoded offline. No compression or extra training is applied.

3.1 Query Vocabulary Usage Exhibits Long-tail Sparsity

Among 50,000 held-out training queries, only 40,546 token IDs (26.7% of the vocabulary) appear. The most frequent 1,000 and 10,000 tokens account for 64.5% and 92.7% of occurrences. BEIR evaluation queries use only 7,450 IDs, with the top 1,000 accounting for 70.8%. Figure 2(a–b) shows this long tail and the potential to reduce EmbeddingBag storage.

The held-out training split reflects the original LightRetriever query distribution, while the BEIR counts provide a separate check on evaluation queries. Both show that the full tokenizer vocabulary is much larger than the subset typically activated by retrieval queries.

3.2 Frequency Does Not Determine Retrieval Importance

Frequency alone does not identify which tokens matter for ranking. We measure each token's positive-rank impact as the relevant document's ranking improvement when that token is present. Figure 2(c) shows wide impact variation among rare tokens: some carry decisive entities, numbers, or domain terms. Deleting rows by frequency can therefore remove retrieval evidence.

Thus, vocabulary sparsity does not imply that every infrequent token is disposable. A token may appear in few queries yet distinguish the correct document from near matches in those queries; a safe compression rule must account for that retrieval role.

3.3 Implications for Query Vocabulary Compression

The observations motivate three design choices: select anchors by retrieval utility, protect critical rare tokens, and fold removed tokens into retained anchors so their contributions remain represented.

4 METHOD

4.1 OVERVIEW

RAVF targets the remaining online bottleneck in asymmetric LLM retrieval: the full-tokenizer query EmbeddingBag. In LightRetriever-style retrieval, documents are encoded offline by an LLM encoder, while queries are represented online through token-level lookup in a non-contextual EmbeddingBag. Although this design removes query-time LLM inference, it still maintains one query embedding row for every tokenizer entry. As shown in Section 3, the query vocabulary contains substantial redundancy, while naive deletion may remove rare but token-level retrieval evidence.

RAVF addresses this problem through retrieval-aware vocabulary transformation. It first identifies retrieval-useful tokens as anchors, then protects token-level retrieval evidence, and finally maps removed tokens to retained anchors instead of discarding them. The document encoder, document descriptors, tokenizer, and retrieval index remain unchanged; only the query-side EmbeddingBag and token-to-anchor mapping are modified.

Because RAVF only modifies the query-side EmbeddingBag, it is compatible with existing document indexes and hybrid retrieval pipelines.

4.2 ASYMMETRIC RETRIEVAL SETUP

Let $\mathcal{V}$ denote the tokenizer vocabulary and $E \in \mathbb{R}^{|\mathcal{V}| \times h}$ the original query EmbeddingBag. Given a tokenized query $x_q = (t_1, \ldots, t_m)$, the non-contextual query descriptor is computed as:

$$\mathbf{q} = f_q(x_q; E) = \mathrm{norm}\left(\frac{1}{m}\sum_{i=1}^{m} E[t_i]\right). \tag{1}$$

The document descriptor $\mathbf{d} = f_d(x_d)$ is generated offline by the frozen document-side LLM encoder, and retrieval is performed using inner product similarity:

$$s(q, d) = \mathbf{q}^\top \mathbf{d}. \tag{2}$$

The goal of RAVF is to replace E with a compact EmbeddingBag

$$\widetilde{E} \in \mathbb{R}^{|\mathcal{A}| \times h}, \tag{3}$$

where $\mathcal{A}$ is the selected anchor vocabulary and $|\mathcal{A}| \ll |\mathcal{V}|$. A token-to-anchor mapping $a : \mathcal{V} \to \{1, \ldots, |\mathcal{A}|\}$ maps original tokens to compact rows. The compressed query representation becomes:

$$\widetilde{\mathbf{q}} = \mathrm{norm}\left(\frac{1}{m}\sum_{i=1}^{m} \widetilde{E}[a(t_i)]\right). \tag{4}$$

This preserves the original retrieval interface because document vectors remain unchanged.

4.3 RETRIEVAL-AWARE TOKEN UTILITY

The first step of RAVF is to select query tokens that serve as anchors. Let $\mathcal{V}_{obs} \subseteq \mathcal{V}$ denote the observed query tokens. Instead of relying only on frequency, RAVF assigns each token a retrieval-aware utility score:

$$u(t) = \lambda_f \bar{c}(t) + \lambda_r \bar{r}(t), \quad t \in \mathcal{V}_{obs}, \tag{5}$$

where $\bar{c}(t)$ is normalized query-token frequency and $\bar{r}(t)$ is the retrieval contribution estimated from frozen teacher rankings. The default weights are $\lambda_f = 0.25$ and $\lambda_r = 0.75$. Although frequency alone is insufficient for safe pruning, it provides a useful prior for estimating token availability. Therefore, RAVF combines frequency with retrieval impact rather than relying on frequency alone.

For each training query, the ranking contribution is computed from teacher retrieval results and qrels positives:

$$r_q = \max(0, s_T(q, d^+) - s_T(q, d^-_{\mathrm{top}})) + \frac{1}{1 + \mathrm{rank}_T(d^+)}, \tag{6}$$

where d^-_{top} denotes the highest-ranked non-positive document. The token-level ranking score $r(t)$ aggregates r_q over queries containing token t. The first term measures the separation between positive and competing documents, while the second term captures the teacher ranking position.

Anchor budget. Given a target retained ratio ρ, RAVF first selects

$$B_\rho = \lceil \rho |\mathcal{V}_{obs}| \rceil \tag{7}$$

tokens with the highest utility scores as candidate anchors.

4.4 Retrieval-Critical Token Protection

Utility ranking alone may remove rare but important retrieval evidence. Therefore, RAVF introduces a protected token set:

$$\mathcal{P} = \mathcal{P}_{special} \cup \mathcal{P}_{lex} \cup \mathcal{P}_{rare} \cup \mathcal{P}_{util}. \tag{8}$$

$\mathcal{P}_{special}$ preserves tokenizer-required tokens, $\mathcal{P}_{lex}$ protects lexical-sensitive patterns such as numbers and entity fragments, and $\mathcal{P}_{rare}$ preserves rare tokens with high retrieval utility:

$$\mathcal{P}_{rare} = \mathrm{Top}_{B_{rare}}\{t \in \mathcal{V}_{obs} : c(t) \le c_{\max}; u(t)\}. \tag{9}$$

The optional utility protection set is defined as:

$$\mathcal{P}_{util} = \{t \in \mathcal{V}_{obs} : u(t) \ge Q_\alpha(u)\}, \tag{10}$$

where Q_α denotes the α-quantile of utility scores.While anchor selection keeps the highest-utility tokens under the retention budget, utility protection preserves a small set of high-utility tokens that would otherwise be excluded under aggressive compression.

The final anchor vocabulary is:

$$\mathcal{A} = \mathrm{Top}_{B_\rho}(\mathcal{V}_{obs}; u) \cup \mathcal{P}. \tag{11}$$

Because protected tokens are added after budget selection, the actual compact vocabulary size is determined by $|\mathcal{A}|$ rather than only by the nominal retained ratio ρ.

4.5 Vocabulary Folding

Unlike pruning methods that discard non-selected tokens, RAVF preserves their contribution through anchor-based folding.

For a removed token $t \in \mathcal{V}_{obs} \setminus \mathcal{A}$, the default mapping is:

$$a(t) = \arg\max_{a \in \mathcal{A}} \cos(E[t], E[a]). \tag{12}$$

For retained anchors, the mapping is identity. During inference, each query token is first mapped through $a(\cdot)$ and then retrieved from the compact EmbeddingBag. Therefore, removed tokens still contribute through their closest retained anchors instead of being discarded completely.

Initialization. Each compact anchor embedding is initialized by combining its original embedding with the embeddings of folded tokens:

$$\widetilde{E}[a] = (1-\beta)E[a] + \beta \frac{1}{|\mathcal{F}(a)|} \sum_{t \in \mathcal{F}(a)} E[t], \tag{13}$$

where $\mathcal{F}(a) = \{t : a(t) = a, t \ne a\}$. This initialization allows anchors to preserve part of the semantic information from folded tokens before calibration.

4.6 Compact EmbeddingBag Tuning

After folding, RAVF calibrates the compact query EmbeddingBag while keeping the document space fixed. For a query q_i, positive document d_i^+, and negative set $\mathcal{G}_i$, we optimize the LightRetriever-style InfoNCE objective:

$$\mathcal{L}_{\mathrm{InfoNCE}} = -\log \frac{\exp(\widetilde{\mathbf{q}}_i^\top \mathbf{d}_i^+ / \tau)}{\sum_{d \in \mathcal{G}_i} \exp(\widetilde{\mathbf{q}}_i^\top \mathbf{d}/\tau)}. \tag{14}$$

The default configuration uses $n = 8$ passages per query and temperature $\tau = 0.02$. Only $\widetilde{E}$ is updated, while the document encoder, document descriptors, tokenizer, and token-to-anchor map remain fixed.

4.7 Deployment and Compatibility

At deployment time, RAVF requires only the compact EmbeddingBag $\widetilde{E}$ and the token-to-anchor map $a(t)$. Since the document encoder and descriptors remain unchanged, existing document indexes can be directly reused. In hybrid retrieval, RAVF only modifies the asymmetric dense query branch, while the sparse branch and score fusion remain.

5 Experiments

5.1 Experimental Setup

We follow the evaluation style of LightRetriever (Ma et al., 2026). The main benchmark setting uses 15 English BEIR (Thakur et al., 2021) tasks and 8 CMTEB (Xiao et al., 2024) Retrieval For efficient ablations, we additionally report results on a fixed six-task BEIR subset (BEIR-6): TRECCOVID, FiQA2018, SCIDOCS, ArguAna, SciFact, and NFCorpus. The exact task set for each average is specified in the corresponding table.

The evaluated architecture is the asymmetric LightRetriever setting: the document-side LLM encoder and document descriptors are fixed, while the query-side EmbeddingBag is compressed. Unless otherwise specified, RAVFuses single-anchor vocabulary folding, token protection, and ordinary InfoNCE tuning.

For training, we use the same preprocessed LightRetriever training stream specified by the original domain configuration. The pilot runs materialize 50K training queries, each with one positive passage and hard negatives, and tune only the compact query EmbeddingBag for 2K steps with batch size 128, seven negatives per query, learning rate 2×10^{-4}, and InfoNCE temperature 0.02. Full-scale runs follow the 12K-step setting used in the final protocol. All compressed variants reuse the same pre-encoded document descriptors and therefore do not require corpus re-encoding.

5.2 Main Benchmark Comparison

Table 1 compares RAVF with lexical baselines, existing retrievers, full LLM retrievers, and LightRetriever. We follow the LightRetriever evaluation protocol and report both retrieval quality and online query-side efficiency. Small annotations report compression, speedup, and nDCG@10 drop relative to the corresponding LightRetriever-Llama3.1-8B baseline of the same retrieval type.

RAVF greatly reduces the query-side cost while keeping most of the retrieval quality. In the hybrid setting, RAVF-Llama8B-0.9 reduces query encoder parameters from 525.3M to 131.1M, giving $4.01\times$ compression and $3.76\times$ speedup, with only 1.3 nDCG@10 drop on BEIR and 0.8 drop on CMTEB-R.

The dense-only setting shows the same trend. RAVF-Llama8B-0.9-Dense achieves $4.01\times$ compression and $4.91\times$ speedup, with only 1.6 and 0.8 nDCG@10 drops on BEIR and CMTEB-R. At the 0.5 setting, it achieves $6.32\times$ compression and $5.54\times$ speedup, while keeping 49.7 BEIR nDCG@10 and 59.1 CMTEB-R nDCG@10. These results show that RAVF improves the accuracy–memory–latency trade-off without changing the document encoder or cached document index.

5.3 Model-Scale Query-Side Efficiency

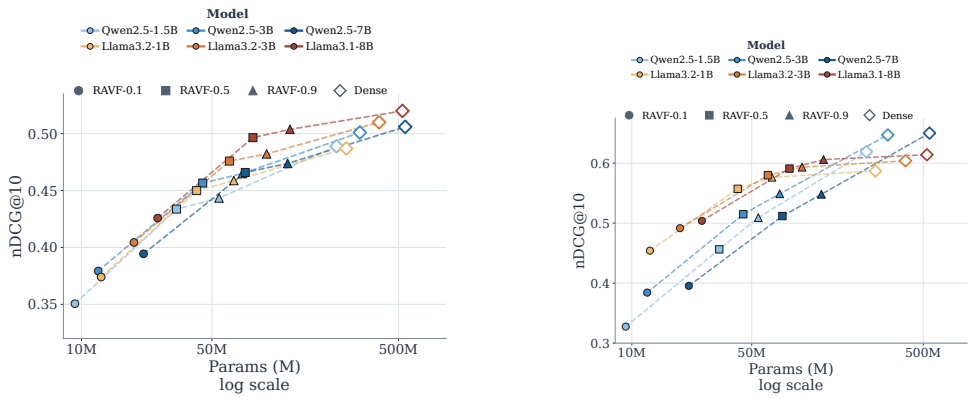
We evaluate whether RAVF generalizes across backbone families and model scales. Figure 3 reports the accuracy–memory trade-off for Qwen2.5 and Llama retrievers from 1B to 8B parameters, where the x-axis denotes query-side EmbeddingBag parameters on a logarithmic scale.

Across both BEIR and CMTEB-R, RAVF consistently reduces query-side parameters while maintaining competitive nDCG@10. For example, on Llama3.1-8B, RAVF-0.9 reduces the query EmbeddingBag from 525.3M to 131.1M parameters, with only a small drop from the dense baseline. This trend holds across both model families and scales.

Table 2 further reports FiQA2018 efficiency under the same dense retrieval setting. All variants use cached document descriptors and the same hardware configuration. RAVF reduces the query EmbeddingBag from hundreds of millions of parameters to tens of millions and consistently lowers

Table 1: Comparison of retrieval effectiveness and efficiency. 'Query+Retr.' reports online query encoding plus retrieval latency on FiQA2018, measured on a single RTX 4090 with `EVAL_BATCH_SIZE=64` and cached document embeddings. Small annotations in RAVF rows report parameter compression and latency speedup over the matching † baseline, while red annotations under nDCG@10 show the absolute nDCG drop; Hybrid rows are compared with the Hybrid baseline and Dense rows with the Dense baseline.

Method	Efficiency		BEIR (15 datasets)				CMTEB-R (8 datasets)			
	Query Encoder Params	Query+Retr. (s)	nDCG@10	R@20	R@50	R@100	nDCG@10	R@20	R@50	R@100
Static Embed (Aarsen, 2025)	31.3M	3.35s	34.1	44.4	53.1	59.5	31.3	45.5	54.3	60.4
BM25 (Robertson et al., 1994)	NN Free	2.87s	41.7	48.8	56.5	61.8	50.8	63.9	70.0	74.1
Static Embed + BM25 (Aarsen, 2025; Robertson et al., 1994)	31.3M	3.75s	44.7	51.9	59.8	65.1	52.1	65.8	71.9	76.1
USTAD (Kim et al., 2024)	66M	4.15s	44.2	–	–	65.3	–	–	–	–
BGE-m3 dense+sparse (Chen et al., 2024)	568M	31.5s	49.6	56.6	63.6	68.8	65.6	79.7	85.2	88.4
SPLADE-v3 (Lassance et al., 2024)	435M	29.5s	50.2	56.6	63.3	68.6	–	–	–	–
LLM2Vec-Llama8B (BehnamGhader et al., 2024)	8B	192.54s	56.6	62.8	69.8	74.9	54.4	67.0	72.3	75.6
E5-Mistral7B (Wang et al., 2024)	7B	181.25s	56.9	62.1	68.9	73.6	61.8	75.1	81.6	87.1
Full-Llama3.1-8B (Grattafiori et al., 2024)	8B	196.71s	56.8	64.3	71.1	75.7	67.6	82.0	87.4	90.9
Full-Qwen2.5-7B (Yang et al., 2024)	7B	184.35s	56.6	63.6	70.5	75.2	70.1	84.4	89.4	92.4
LightRetriever-Llama3.1-8B-Hybrid(dense+sparse)† (Ma et al., 2026)	525.3M	24.96s	54.4	60.9	67.7	72.8	63.0	77.3	83.6	87.4
LightRetriever-Qwen2.5-7B-Hybrid(dense+sparse) (Ma et al., 2026)	543.6M	31.25s	53.8	60.3	67.5	72.5	66.5	81.1	86.7	89.7
LightRetriever-Llama3.1-8B-Dense† (Ma et al., 2026)	525.3M	21.59s	52.0	–	–	–	61.4	–	–	–
LightRetriever-Qwen2.5-7B-Dense (Ma et al., 2026)	543.6M	24.22s	50.6	–	–	–	65.0	–	–	–
RAVF-Llama8B-0.9-Hybrid(dense+sparse)	131.1M$_{/4.01}$	6.64s$_{\times 3.76}$	53.1$_{-1.3}$	60.3	66.9	71.9	62.2$_{-0.8}$	76.7	82.8	86.5
RAVF-Llama8B-0.5-Hybrid(dense+sparse)	83.1M$_{/6.32}$	6.18s$_{\times 4.04}$	52.7$_{-1.7}$	59.6	66.3	71.5	61.2$_{-1.8}$	76.0	82.2	86.1
RAVF-Llama8B-0.1-Hybrid(dense+sparse)	25.6M$_{/20.52}$	6.04s$_{\times 4.13}$	49.5$_{-4.9}$	55.0	62.4	68.4	56.6$_{-6.4}$	71.4	78.3	83.0
RAVF-Llama8B-0.9-Dense	131.1M$_{/4.01}$	4.40s$_{\times 4.91}$	50.4$_{-1.6}$	56.4	63.1	67.9	60.6$_{-0.8}$	75.1	81.5	85.6
RAVF-Llama8B-0.5-Dense	83.1M$_{/6.32}$	3.90s$_{\times 5.54}$	49.7$_{-2.3}$	55.9	62.7	67.4	59.1$_{-2.3}$	73.7	80.6	84.8
RAVF-Llama8B-0.1-Dense	25.6M$_{/20.52}$	3.49s$_{\times 6.19}$	42.6$_{-9.4}$	50.2	57.0	61.9	50.4$_{-11.0}$	65.3	72.9	78.3



(a) BEIR(Full) nDCG@10.

(b) CMTEB-R(Full) nDCG@10.

Figure 3: Accuracy–memory trade-off across model scales. The x-axis shows query-side EmbeddingBag parameters in log scale, and the y-axis shows nDCG@10.

online query-plus-retrieval time, showing that its query-side efficiency gain is not tied to a single backbone.

5.4 Ablation Study

Table 3 isolates the main design choices in the query vocabulary interface. We evaluate three retained vocabulary ratios and report nDCG@10 and Recall@100 side by side for each ratio. The comparison tests on BEIR-6. For context, we also include the corresponding LightRetriever dense and hybrid nDCG@10 values from the original paper (Ma et al., 2026), averaged over the same six tasks; the original paper does not report Recall@100 for these rows.

The ablation shows that direct deletion is the main source of degradation. At 10% retained vocabulary, utility deletion reaches only 0.2540 nDCG@10, whereas folding raises it to 0.3942. Token protection further improves the result to 0.4181, indicating that fragile token-level retrieval evidence is not fully captured by the global utility score. InfoNCE tuning adds a smaller but consistent calibration gain, reaching 0.4264 nDCG@10 and 0.5484 Recall@100. At this retention level, folding contributes the largest nDCG@10 increase over utility deletion (+0.1402), followed by protection

Table 2: Query-side EmbeddingBag size and FiQA2018 retrieval efficiency across model scales. All document descriptors are cached. All methods in this table use Dense retrieval. Params = Dim × #Vecs.

Model	Dim	Lightretriever-Dense			RAVF-Retain 0.9			RAVF-Retain 0.5			RAVF-Retain 0.1		
		#Vecs	Params	Time	#Vecs	Params	Time	#Vecs	Params	Time	#Vecs	Params	Time
Qwen2.5-7B	3584	151666	543.6M	24.22	35554	127.4M	5.22	21062	75.5M	3.52	6000	21.5M	3.27
Qwen2.5-3B	2048	151666	310.6M	16.96	35555	72.8M	2.99	21802	44.7M	2.71	6001	12.3M	2.47
Qwen2.5-1.5B	1536	151666	233.0M	8.89†	35554	54.6M	3.08	21045	32.3M	2.87	6000	9.2M	2.26
Llama3.1-8B	4096	128256	525.3M	21.59	32014	131.1M	4.40	20266	83.1M	3.90	6251	25.6M	3.49
Llama3.2-3B	3072	128256	394.0M	17.26	32000	98.3M	3.67	20226	62.1M	3.41	6219	19.1M	3.04
Llama3.2-1B	2048	128256	262.7M	10.41	32010	65.6M	3.15	20236	41.4M	2.94	6227	12.8M	2.67

Table 3: Core ablation on the BEIR-6 with Llama3.1-8B.

Method	Retain 0.9		Retain 0.5		Retain 0.1	
	nDCG	R@100	nDCG	R@100	nDCG	R@100
Lightretriever Dense	0.4883	–	0.4883	–	0.4883	–
Random del.	0.4569	0.5650	0.2931	0.4625	0.0507	0.1661
Freq. del.	0.4804	0.5785	0.4485	0.5619	0.2460	0.4222
Utility del.	0.4794	0.5771	0.4495	0.5608	0.2540	0.4295
Util. + Fold (A)	0.4842	0.5815	0.4767	0.5782	0.3942	0.5311
A + Protect (B)	0.4836	0.5815	0.4778	0.5788	0.4181	0.5442
B + InfoNCE (RAVF)	**0.4863**	**0.5828**	**0.4802**	**0.5805**	**0.4264**	**0.5484**

(+0.0239) and tuning (+0.0083). These increments distinguish the effect of preserving token contributions from the later calibration of their compact representations.

5.5 Rank Recovery Analysis

We further test whether RAVF recovers ranking evidence lost by direct deletion. Using Llama3.1-8B on BEIR-6, we track the rank of the best positive document under FullBag, direct deletion, and RAVF for queries whose positive documents are harmed by deletion. Figure 4 shows that deletion substantially harms positive-document ranks, while RAVF restores them toward FullBag. At retain 0.5, deletion damages 30.45% of queries and increases the mean positive rank from 52.66 to 69.65; RAVF reduces it to 56.84. At retain 0.1, deletion damages 64.37% of queries and increases the mean rank to 162.28, while RAVF reduces it to 70.85. These results indicate that folding preserves useful token contributions that would otherwise be removed by direct pruning. This rank analysis considers only queries harmed by deletion, so it diagnoses that failure mode rather than estimating performance over all queries. The full-set results in Table 4 provide complementary evidence that folding outperforms deletion-based alternatives across retained vocabulary ratios.

To further understand this behavior, Figure 5 provides qualitative examples where direct deletion removes rare but retrieval-critical tokens. RAVF preserves their contribution through protection and folding. These results show that RAVF improves retrieval by recovering damaged ranking evidence rather than only smoothing aggregate metrics, supporting vocabulary folding as an evidence-preserving compression strategy.

5.6 Pruning Method Comparison

Table 4 compares RAVF with common deletion-based vocabulary pruning criteria. Random, frequency, and IDF pruning use token statistics, while magnitude and Taylor pruning use embedding parameters or loss sensitivity. Deletion-based pruning degrades rapidly under aggressive compression. At retain 0.1, frequency deletion reaches 0.2460 nDCG@10 and Taylor deletion reaches 0.1861, while RAVF achieves 0.4181. This gap shows that selecting better tokens is not sufficient when removed tokens are discarded.

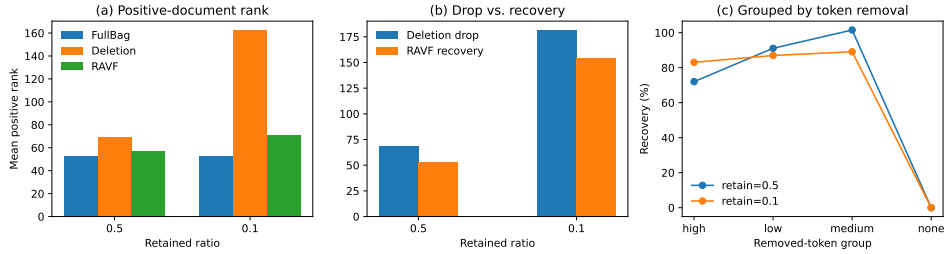


Figure 4: Rank recovery analysis with Llama3.1-8B on BEIR-6. Lower positive-document rank is better. Direct deletion pushes relevant documents to worse ranks, while RAVF recovers most of the deletion-induced rank drop.

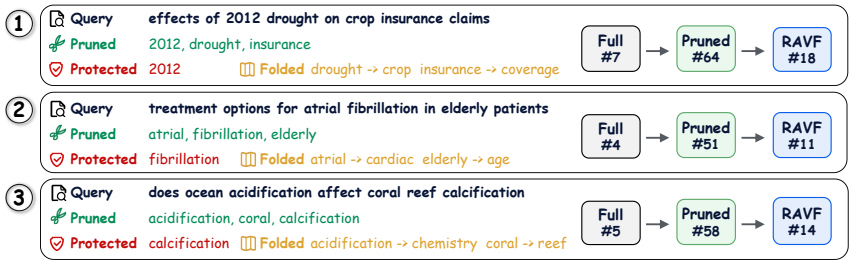


Figure 5: Qualitative examples of ranking evidence recovery with Llama3.1-8B under retain ratio 0.5. Examples are sampled from FiQA2018, NFCorpus, and SciFact queries in BEIR where deletion induces a rank drop at a 0.5 retain ratio. Utility-based deletion removes selected tokens, while RAVF preserves their contribution through protection and folding.

Overall, RAVF performs best among methods across all ratios. The result supports our central claim: effective query vocabulary compression should preserve retrieval evidence through folding rather than only deleting low-priority tokens.

Table 4: Comparison with deletion-based vocabulary pruning methods on BEIR-6 with Llama3.1-8B. All methods are evaluated without additional training.

Method	Retain 0.9		Retain 0.5		Retain 0.1	
	nDCG	R@100	nDCG	R@100	nDCG	R@100
LightRetriever-Dense	0.4883	–	0.4883	–	0.4883	–
Random deletion	0.4569	0.5651	0.2931	0.4625	0.0504	0.1654
Frequency deletion	0.4804	0.5785	0.4485	0.5619	0.2460	0.4219
IDF deletion	0.3396	0.4868	0.0950	0.2408	0.0133	0.0509
Magnitude deletion	0.4501	0.5633	0.2610	0.4327	0.0363	0.1377
Taylor deletion	0.4811	0.5806	0.4234	0.5508	0.1861	0.3491
RAVF (without tuning)	**0.4836**	**0.5815**	**0.4778**	**0.5788**	**0.4181**	**0.5442**

6 Conclusion

We studied query-side vocabulary compression for asymmetric LLM retrieval. Our analysis shows that the query EmbeddingBag is highly redundant, but naive deletion is unsafe because rare tokens may carry critical retrieval evidence. We propose RAVF, which preserves such evidence through retrieval-aware anchor selection, token protection, and vocabulary folding. Experiments across BEIR, CMTEB-R, and multiple backbone scales demonstrate that RAVF improves the accuracy-efficiency trade-off while remaining compatible with hybrid retrieval. The results without additional tuning show that folding itself retains ranking evidence under aggressive compression. Optional calibration of the compact EmbeddingBag provides a further adjustment while keeping document descriptors and the index fixed. Future work will explore whether retrieval-aware folding can be extended to other tasks with large discrete vocabularies and token-based representations.

AI USE STATEMENT

In this work, we used generative AI tools sparingly to refine the writing, locate potentially relevant publications, and support early-stage research planning and execution. We have not used generative AI tools for other tasks requiring disclosure, and the remaining required disclosure categories are not applicable to this work. Additionally, we have not used generative AI tools for other tasks with recommended disclosure. We have reviewed all AI-assisted work, verified the relevant publications against their original sources, and independently assessed the AI-assisted research work. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Tom Aarsen. Train 400x faster static embedding models with sentence transformers, January 2025. URL https://huggingface.co/blog/static-embeddings. Hugging Face Blog.

Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. LLM2Vec: Large language models are secretly powerful text encoders. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=IW1PR7vEBf.

Jianlyu Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. M3-embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, volume ACL 2024 of *Findings of ACL*, pp. 2318–2335. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.FINDINGS-ACL.137. URL https://doi.org/10.18653/v1/2024.findings-acl.137.

Aaron Grattafiori et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. doi: 10.48550/arXiv.2407.21783. URL https://arxiv.org/abs/2407.21783.

Song Han, Huizi Mao, and William J. Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. In *International Conference on Learning Representations*, 2016.

Karen Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1):11–21, 1972. doi: 10.1108/EB026526.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6769–6781, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.550. URL https://aclanthology.org/2020.emnlp-main.550.

Omar Khattab and Matei Zaharia. ColBERT: Efficient and effective passage search via contextualized late interaction over BERT. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 39–48, 2020. doi: 10.1145/3397271.3401075.

Seungyeon Kim, Ankit Singh Rawat, Manzil Zaheer, Wittawat Jitkrittum, Veeranjaneyulu Sadhanala, Sadeep Jayasumana, Aditya Krishna Menon, Rob Fergus, and Sanjiv Kumar. USTAD: unified single-model training achieving diverse scores for information retrieval. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=LbEB39lZqp.

Carlos Lassance, Hervé Déjean, Thibault Formal, and Stéphane Clinchant. SPLADE-v3: New baselines for SPLADE. *CoRR*, abs/2403.06789, 2024. doi: 10.48550/ARXIV.2403.06789. URL https://doi.org/10.48550/arXiv.2403.06789.

Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. NV-Embed: Improved techniques for training LLMs as generalist embedding models. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=lgsyLSsDRe.

Guangyuan Ma, Yongliang Ma, Xuanrui Gou, Zhenpeng Su, Ming Zhou, and Songlin Hu. LightRetriever: A LLM-based text retrieval architecture with extremely faster query inference. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=vNEY32I8Y8.

Pavlo Molchanov, Stephen Tyree, Tero Karras, Timo Aila, and Jan Kautz. Pruning convolutional neural networks for resource efficient inference. In *International Conference on Learning Representations*, 2017.

Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. MTEB: Massive text embedding benchmark. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 2014–2037, 2023. doi: 10.18653/v1/2023.eacl-main.148. URL https://aclanthology.org/2023.eacl-main.148/.

Stephen E. Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. Okapi at TREC-3. In Donna K. Harman (ed.), *Proceedings of The Third Text REtrieval Conference, TREC 1994, Gaithersburg, Maryland, USA, November 2-4, 1994*, volume 500-225 of *NIST Special Publication*, pp. 109–126. National Institute of Standards and Technology (NIST), 1994. URL http://trec.nist.gov/pubs/trec3/papers/city.ps.gz.

Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. ColBERTv2: Effective and efficient retrieval via lightweight late interaction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 3715–3734, 2022. doi: 10.18653/v1/2022.naacl-main.272. URL https://aclanthology.org/2022.naacl-main.272/.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. *NeurIPS Datasets and Benchmarks*, 2021.

Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748, 2018. URL http://arxiv.org/abs/1807.03748.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. Improving text embeddings with large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pp. 11897–11916. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.ACL-LONG.642. URL https://doi.org/10.18653/v1/2024.acl-long.642.

Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. C-pack: Packed resources for general chinese embeddings. In Grace Hui Yang, Hongning Wang, Sam Han, Claudia Hauff, Guido Zuccon, and Yi Zhang (eds.), *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2024, Washington DC, USA, July 14-18, 2024*, pp. 641–649. ACM, 2024. doi: 10.1145/3626772.3657878. URL https://doi.org/10.1145/3626772.3657878.

An Yang et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024. doi: 10.48550/arXiv.2412.15115. URL https://arxiv.org/abs/2412.15115.

A Appendix

A.1 Additional Experimental Details

Training Protocol and Frozen Components. Unless otherwise stated, all RAVF experiments keep the backbone LLM encoder and document representations frozen. Training only updates the compact query-side EmbeddingBag after vocabulary folding.

Benchmark Definition and Baseline Reporting. Throughout the appendix, BEIR-6 refers to TRECCOVID, FiQA2018, SCIDOCS, ArguAna, SciFact, and NFCorpus. Results marked as Lightretriever, Full Model, and prior retrievers are copied from Ma et al. (2025) when available. RAVF rows are produced by our implementation under the same task definitions. When a metric is not reported in the original paper, we mark it as unavailable rather than recomputing it with a different protocol.

Training Data Reuse and Evaluation Isolation. All training pools are constructed once with a fixed random seed from the original LightRetriever training stream. The same materialized query-positive-negative triples, frozen document descriptors, and retrieval caches are reused across retained ratios, objectives, and seeds unless explicitly stated. No BEIR or CMTEB-R evaluation queries or qrels are used for vocabulary construction, anchor selection, folding, or tuning.

Parameter accounting. The reported query-side parameters only include trainable parameters of the dense query encoder. In the hybrid setting, the sparse branch is based on tokenization and lexical matching rather than an additional neural encoder. Therefore, it introduces negligible learnable parameters (e.g., the tokenizer vocabulary requires only around 0.15M parameters for Qwen2.5), which are omitted from the parameter accounting. Since dense and hybrid LightRetriever share the same query EmbeddingBag, their reported query-side parameter sizes are identical.

Effective retention ratio. The retain ratio in RAVF controls the initial anchor selection budget rather than the final vocabulary size. The final compact vocabulary additionally includes protected tokens and tokenizer-required entries:

$$|\mathcal{A}| = \mathrm{Top}_{B_\rho}(\mathcal{V}_{obs}; u) \cup \mathcal{P}.$$

Moreover, RAVF operates on the observed query vocabulary instead of the full tokenizer vocabulary, since unused tokens do not contribute to query representations. Therefore, the final parameter reduction is not expected to scale linearly with the nominal retain ratio. Under aggressive compression, the protected token set becomes a larger fraction of the remaining vocabulary, leading to a smaller-than-linear reduction.

A.2 Implementation and Hardware

Experiments are implemented in PyTorch 2.3.0 with Transformers 4.47.1, PEFT 0.13.2, Datasets 2.21.0, MTEB 1.14.15, and FAISS 1.7.4/1.8 depending on CPU/GPU indexing. Training and large-scale encoding are run on NVIDIA A800/RTX 4090 GPUs. Latency is measured on a single RTX 4090 with cached document descriptors, batch size 64, and excludes one-time document encoding and full EmbeddingBag construction.

A.3 Training Data Details

RAVF follows the LightRetriever training stream rather than evaluation queries. The preprocessed mixture contains 20 English datasets and 3 Chinese datasets with 8.38M retrieval examples (Ma et al., 2026). Each example contains one query, one positive passage, and hard negatives. We use the same training format and only update the compact query EmbeddingBag; evaluation queries are never used for vocabulary construction or tuning.

A.4 Evaluation Dataset Details

The main English benchmark is BEIR (Thakur et al., 2021). Following LightRetriever, the full evaluation uses 15 BEIR retrieval tasks: ArguAna, ClimateFEVER, CQADupStack, DBPedia,

FEVER, FiQA2018, HotpotQA, MSMARCO, NFCorpus, NQ, QuoraRetrieval, SCIDOCS, SciFact, Touche2020, and TRECCOVID. MSMARCO uses the dev split; the other tasks use their standard test splits. Table 5 shows the task grouping used in the paper.

Table 5: Recommended BEIR task grouping for the full evaluation. Exact task names should follow the installed MTEB version.

Task group	Tasks
Open-domain QA	NQ, HotpotQA, MSMARCO
Fact verification	FEVER, ClimateFEVER, SciFact
Domain retrieval	FiQA2018, NFCorpus, SCIDOCS, TRECCOVID
Argument/entity retrieval	ArguAna, DBPedia, Touche2020
Duplicate retrieval	QuoraRetrieval, CQADupStack variants

The Chinese extension follows the retrieval subset used by LightRetriever in CMTEB/MTEB-style evaluation (Muennighoff et al., 2023). It contains 8 tasks: CmedqaRetrieval, CovidRetrieval, DuRetrieval, EcomRetrieval, MMarcoRetrieval, MedicalRetrieval, T2Retrieval, and VideoRetrieval. We keep these results in the appendix or extended evaluation unless all English experiments are completed, because the main claim of this paper is query vocabulary folding under an asymmetric retriever rather than multilingual benchmark leadership.

Table 6: Chinese retrieval tasks for extended evaluation, aligned with LightRetriever's CMTEB Retrieval protocol.

Task group	Tasks
Medical and health	CmedqaRetrieval, MedicalRetrieval, CovidRetrieval
Web and passage search	MMarcoRetrieval, T2Retrieval, DuRetrieval
Commerce and media	EcomRetrieval, VideoRetrieval

The current draft uses fixed small BEIR subsets for rapid method selection. For the core ablation below, we use TRECCOVID, FiQA2018, SCIDOCS, ArguAna, SciFact, and NFCorpus, covering biomedical retrieval, financial QA, citation search, argument retrieval, scientific fact verification, and domain-specific retrieval. BEIR-6 is used only for controlled ablations due to the cost of repeatedly evaluating large LLM-based retrievers. It covers biomedical retrieval, finance, citation search, argument retrieval, scientific fact verification, and domain-specific retrieval. Final benchmark comparisons are reported on the full BEIR and CMTEB-R suites. We reserve the full BEIR suite for the final benchmark table.

Table 7: Effect of rare-token protection under extreme vocabulary compression on BEIR-6 with Llama3.1-8B. Each entry reports the actual retained vocabulary size and nDCG@10.

Rare Protected Tokens	Retain Ratio (actual rows / nDCG@10)		
	0.1	0.01	0.001
4096	8206 / 0.4337	5058 / 0.3607	4749 / 0.3340
2048	6251 / 0.4181	3103 / 0.3090	2794 / 0.2606
1024	5246 / 0.4099	2098 / 0.2516	1789 / 0.1825
512	4742 / 0.4046	1594 / 0.2007	1285 / 0.1237
256	4489 / 0.3988	1341 / 0.1613	1032 / 0.0771
128	4361 / 0.3989	1213 / 0.1322	904 / 0.0525

A.5 Effect of Training Steps

Table 8 studies the effect of calibration length. At retain 0.1, tuning consistently improves over the zero-step folded model and largely saturates after 2K–4K steps. At retain 0.5, shorter calibration is sufficient, while longer training provides limited additional gains, suggesting that the compact query interface can be calibrated with a small number of updates. To balance the training time and effect, we chose 2k steps in the main paper.

Table 8: Effect of InfoNCE training steps on BEIR-6 with Llama3.1-8B. All runs use a fixed 50K-query training pool and seed 13. Zero steps denotes RAVF folding and protection without additional tuning.

Steps	Retain 0.5		Retain 0.1	
	nDCG	R@100	nDCG	R@100
0	0.4778	0.5789	0.4181	0.5442
500	0.4804	0.5801	0.4217	0.5462
2K	0.4802	**0.5805**	**0.4264**	0.5484
4K	**0.4808**	0.5798	0.4252	0.5474
8K	0.4762	0.5803	0.4260	**0.5502**
12K	0.4747	0.5796	0.4257	0.5501

Table 9: Effect of training-pool size on BEIR-6 with Llama3.1-8B. All runs use 2K InfoNCE steps and seed 13. The pool is sampled once from the original LightRetriever training stream and reused across settings.

Pool Size	Retain 0.5		Retain 0.1	
	nDCG	R@100	nDCG	R@100
1K	0.4784	0.5805	0.4236	0.5484
5K	0.4788	0.5808	0.4248	0.5479
10K	0.4785	**0.5810**	0.4253	**0.5490**
25K	0.4799	0.5807	0.4262	0.5486
50K	**0.4802**	0.5805	**0.4264**	0.5484

A.6 Effect of Training-Pool Size

RAVF does not require retraining the document encoder or learning a new retriever from scratch. Its training stage only calibrates the compact query-side EmbeddingBag after vocabulary folding. Therefore, we use a fixed and moderate calibration pool rather than the full retrieval training stream. Specifically, the training pool is materialized once from the original LightRetriever training stream with a fixed seed and reused across all retained ratios and variants. Each instance follows the same preprocessing configuration as the main experiments, including one positive passage, sampled negatives, prompt-enabled inputs, and homogeneous batching. No BEIR or CMTEB-R evaluation labels are used when constructing this pool.

To examine whether 50K queries are sufficient, we vary the number of training queries while keeping the evaluation queries, document descriptors, retrieval index, optimization steps, and all other hyperparameters unchanged. Table 9 shows that RAVF already reaches stable performance with relatively small pools. Increasing the pool size from 10K to 50K gives only marginal gains: for retain 0.5, nDCG improves from 0.4785 to 0.4802; for retain 0.1, nDCG improves from 0.4253 to 0.4264. This indicates that the training stage mainly performs query-side calibration rather than large-scale representation learning. We therefore use 50K queries as a conservative default budget: it is large enough to cover diverse query tokens, but avoids unnecessary computation from processing the full training stream.

A.7 Hyperparameter Analysis – Token Protection

To avoid removing rare but retrieval-critical tokens, RAVF introduces a protected token set in addition to utility-based anchor selection. Specifically, RAVF preserves low-frequency tokens with high retrieval utility:

$$\mathcal{P}_{rare} = \mathrm{Top}^{u(t)}_{B_{rare}} \left(\{ t \in \mathcal{V}_{obs} : c(t) \le c_{\max} \} \right). \tag{15}$$

These protected tokens remain as independent anchors during vocabulary folding. We further study the effect of the protection budget under extreme compression by varying B_{rare} while keeping the other components unchanged. Table 7 reports the resulting effective vocabulary size and retrieval performance.

Increasing the protection budget consistently improves nDCG@10, especially at extreme compression ratios (retain ratio 0.01 and 0.001), showing that rare tokens can preserve important retrieval evidence. However, stronger protection also increases the effective vocabulary size, introducing a

Table 10: Utility-weight sensitivity on BEIR-6 with Llama3.1-8B at retain ratio 0.5. All runs use 2K training steps and seed 13. The default setting is $(\lambda_f, \lambda_r) = (0.25, 0.75)$.

λ_f	λ_r	#Anchors	nDCG@10	R@100
0.00	1.00	20,271	0.4791	0.5788
0.10	0.90	20,269	0.4797	0.5795
0.25	0.75	20,266	**0.4802**	0.5805
0.50	0.50	20,261	0.4800	**0.5807**
0.75	0.25	20,254	0.4796	0.5802
1.00	0.00	20,015	0.4786	0.5787

Table 11: Random-seed stability on BEIR-6 with Llama3.1-8B. Seed 13 is used in the main experiments, while Seeds 21 and 42 provide two additional runs. The last two columns report standard deviations across the three seeds.

Retain	Seed 13		Seed 21		Seed 42		Std.	
	nDCG	R@100	nDCG	R@100	nDCG	R@100	nDCG	R@100
0.9	0.4863	0.5828	0.4862	0.5828	0.4865	0.5828	0.0002	0.0000
0.5	0.4802	0.5805	0.4803	0.5800	0.4804	0.5805	0.0001	0.0003
0.1	0.4264	0.5484	0.4266	0.5478	0.4269	0.5485	0.0003	0.0004

trade-off between retrieval accuracy and compression. The default setting of 2048 protected rare tokens provides a balanced operating point between evidence preservation and vocabulary compactness.

A.8 Hyperparameter Analysis – Utility Weight Sensitivity

Table 10 studies the impact of balancing token frequency and retrieval contribution in the utility score $u(t) = \lambda_f \bar{c}(t) + \lambda_r \bar{r}(t)$. We vary λ_f and λ_r while keeping all other components unchanged. Results remain stable across different weight combinations, indicating that RAVF does not rely on a specific weighting choice. The default setting $(\lambda_f, \lambda_r) = (0.25, 0.75)$ achieves competitive performance, while retrieval contribution alone also performs well, confirming that retrieval-aware signals are the main factor for identifying useful anchor tokens.

A.9 Random-Seed Stability

Table 11 reports results across random seeds. The vocabulary construction process is fixed across seeds, and only the calibration initialization changes. The tuned results show only small variation. We additionally report the seed-13 result used in the main paper.

A.10 Training Objective Analysis

Table 12 compares different calibration objectives after applying RAVF compression. We evaluate standard InfoNCE against label-based and teacher-guided objectives, including qrels pairwise supervision, Matrix-KL, teacher-hard InfoNCE, and teacher-weighted InfoNCE.

Let q_θ be the compact query descriptor produced by the trainable folded EmbeddingBag and let d_i be a fixed document descriptor. Ordinary InfoNCE optimizes the query descriptor against one positive and sampled negatives,

$$\mathcal{L}_{\mathrm{NCE}} = -\log \frac{\exp(s(q_\theta, d^+)/\tau)}{\exp(s(q_\theta, d^+)/\tau) + \sum_{d^- \in \mathcal{N}} \exp(s(q_\theta, d^-)/\tau)}. \tag{16}$$

The qrels pairwise objective directly enforces a margin between labeled positives and negatives,

$$\mathcal{L}_{\mathrm{qrels}} = \sum_{d^+, d^-} \log\left(1 + \exp\left[-(s(q_\theta, d^+) - s(q_\theta, d^-))\right]\right). \tag{17}$$

Matrix-KL distillation matches the teacher and student similarity distributions over the same candidate document set $\mathcal{C}$,

$$\mathcal{L}_{\mathrm{KL}} = \mathrm{KL}\left(\mathrm{softmax}(S_T(q, \mathcal{C})/\tau) \parallel \mathrm{softmax}(S_S(q_\theta, \mathcal{C})/\tau)\right). \tag{18}$$

Teacher-hard InfoNCE samples negatives from teacher-ranked hard documents, while teacher-weighted InfoNCE weights negative terms by teacher scores. In all cases, the document descriptors remain fixed; only the compact query EmbeddingBag is updated.

Across retention ratios, stronger teacher-guided objectives do not provide consistent improvements over standard InfoNCE. This suggests that the main benefit of RAVF comes from retrieval-aware token selection and evidence-preserving folding, while lightweight calibration is sufficient to adapt the compact query representation.

Table 12: Comparison of calibration objectives for RAVF under different retention ratios on the six-task BEIR subset with Llama3.1-8B.

Training objective	Retain 0.9		Retain 0.5		Retain 0.1	
	nDCG	R@100	nDCG	R@100	nDCG	R@100
Lightretriever Dense	0.4883	–	0.4883	–	0.4883	–
InfoNCE	0.4863	**0.5828**	0.4802	**0.5805**	0.4264	**0.5484**
Qrels pairwise	0.4789	0.5807	0.4735	0.5787	0.4115	0.5416
Matrix KL	0.4856	0.5827	0.4789	0.5804	0.4233	0.5474
Teacher-hard InfoNCE	0.4860	0.5811	0.4803	0.5788	0.4259	0.5466
Teacher-weighted InfoNCE	0.4840	0.5821	0.4776	0.5788	0.4121	0.5380

A.11 Per-Task Core Ablation

Table 16 expands the core ablation in Table 3 (in main paper) to individual BEIR tasks. nDCG@10 and Recall@100 are placed next to each other for each task. The Dense and Hybrid rows are copied from the original LightRetriever paper and therefore contain only nDCG@10; the original paper does not provide task-level Recall@100 for these two rows.

A.12 Pruning Method Details

This section expands Table 4 (in main paper). Let $\mathcal{Q}$ be the training-query set, x_t be the query EmbeddingBag row for token t, and $\mathcal{A}$ be the retained anchor set. Random deletion assigns each token an independent random score. Frequency pruning keeps tokens with large query occurrence count,

$$s_{\mathrm{freq}}(t) = \sum_{q \in \mathcal{Q}} \mathbf{1}[t \in q]. \tag{19}$$

IDF pruning keeps tokens with high inverse document/query frequency,

$$s_{\mathrm{idf}}(t) = \log \frac{|\mathcal{Q}| + 1}{\mathrm{df}_{\mathcal{Q}}(t) + 1}, \tag{20}$$

where $\mathrm{df}_{\mathcal{Q}}(t)$ counts the number of queries containing t. Magnitude pruning uses the row norm,

$$s_{\mathrm{mag}}(t) = \|x_t\|_2. \tag{21}$$

Taylor pruning estimates first-order loss sensitivity,

$$s_{\mathrm{taylor}}(t) = \left|\langle \nabla_{x_t} \mathcal{L}, x_t \rangle\right|. \tag{22}$$

Our retrieval-aware variant scores tokens by their contribution to retrieval quality and does not simply delete low-scoring rows. Removed tokens are folded into retained anchors,

$$a(t) = \arg\max_{a \in \mathcal{A}} \cos(x_t, x_a), \tag{23}$$

while token protection forces high-utility or fragile lexical tokens to remain explicit. Table 17 reports the per-task results. The main pattern is that generic signals are competitive only under mild compression, while folding becomes essential as the retained vocabulary shrinks.

A.13 Per-task retrieval results

Detailed per-task retrieval results on BEIR and CMTEB-R are provided in Tables 14, 13, and 15.

Table 13: Detailed BEIR nDCG@10 results across individual datasets. RAVF variants are compared with the corresponding LightRetriever dense baseline under the same backbone. The retain ratio denotes the fraction of retained query vocabulary entries.

Method	Params	ArguAna	CQADup	CFEVER	DBPedia	FEVER	FiQA	HotpotQA	MSMARCO
Static Embedding (Aarsen, 2025)	31.3M	44.4	22.0	20.4	27.4	43.0	20.0	46.8	17.9
BM25 (Robertson et al., 1994)	NN Free	31.5	29.9	21.3	31.3	75.3	23.6	60.3	22.8
USTAD (Kim et al., 2024)	66M	34.9	30.6	22.5	35.9	76.9	29.5	56.0	46.6
BGE-M3-Dense (Chen et al., 2024)	568M	53.9	38.4	26.5	38.5	77.2	40.8	68.6	37.6
BGE-M3-Sparse (Chen et al., 2024)	568M	36.2	30.5	26.7	24.4	85.9	26.9	70.3	17.1
SPLADE-v3 (Lassance et al., 2024)	435M	50.9	34.7	23.3	45.0	79.6	37.4	69.2	46.8
LLM2Vec-Llama8B (BehnamGhader et al., 2024)	8B	62.8	48.3	34.3	48.3	90.2	55.3	71.8	43.2
E5-Mistral7B (Wang et al., 2024)	7B	61.9	42.9	38.3	48.9	87.8	56.6	75.7	43.1
Llama3.2-1B									
Full-Llama3.2-1B	1B	58.0	44.0	41.2	43.2	89.2	45.5	69.3	42.7
LightRetriever-Llama3.2-1B-Dense (Ma et al., 2026)	262.7M	49.2	38.1	37.9	34.2	83.6	40.4	60.8	39.2
RAVF-Llama3.2-1B-0.9-Dense	65.6M	49.0	34.3	25.0	30.6	76.1	38.0	57.3	37.1
RAVF-Llama3.2-1B-0.5-Dense	41.4M	48.9	33.7	24.6	29.2	74.2	37.1	55.1	36.0
RAVF-Llama3.2-1B-0.1-Dense	12.8M	43.8	29.4	20.7	16.7	59.2	33.0	40.4	26.4
Llama3.2-3B									
Full-Llama3.2-3B	3B	58.5	46.8	43.7	46.4	89.9	53.4	73.7	44.3
LightRetriever-Llama3.2-3B-Dense (Ma et al., 2026)	394.0M	55.3	40.3	41.7	36.5	84.0	46.2	64.3	41.6
RAVF-Llama3.2-3B-0.9-Dense	98.3M	51.7	37.0	30.4	32.4	77.9	44.9	61.0	39.2
RAVF-Llama3.2-3B-0.5-Dense	62.1M	51.6	36.5	29.9	31.1	76.2	44.4	59.2	38.3
RAVF-Llama3.2-3B-0.1-Dense	19.1M	48.1	32.5	25.4	19.0	62.7	39.0	46.1	29.6
Llama3.1-8B									
Full-Llama3.1-8B	8B	57.1	49.2	44.1	48.2	90.9	59.0	77.8	46.2
LightRetriever-Llama3.1-8B-Dense (Ma et al., 2026)	525.3M	55.8	40.9	41.5	38.7	85.7	49.8	67.6	43.4
LightRetriever-Llama3.1-8B-Hybrid (Ma et al., 2026)	525.3M	57.2	41.9	42.4	42.0	87.8	50.7	71.5	44.5
RAVF-Llama3.1-8B-0.9-Dense	131.1M	55.4	50.6	28.7	32.9	80.8	48.6	64.4	40.6
RAVF-Llama3.1-8B-0.5-Dense	83.1M	55.1	50.1	28.4	31.9	79.5	48.0	62.8	39.7
RAVF-Llama3.1-8B-0.1-Dense	25.6M	52.4	42.9	22.5	19.8	67.4	43.8	49.8	31.2
RAVF-Llama3.1-8B-0.9-Hybrid	131.1M	56.3	39.0	41.1	40.7	86.5	49.8	70.2	43.2
RAVF-Llama3.1-8B-0.5-Hybrid	83.1M	55.9	38.7	40.7	40.3	86.1	49.5	69.8	42.8
RAVF-Llama3.1-8B-0.1-Hybrid	25.6M	54.2	36.3	37.5	37.1	82.9	46.8	66.6	39.6
Qwen2.5-1.5B									
Full-Qwen2.5-1.5B	1.5B	56.6	44.1	40.4	43.9	88.9	47.9	68.1	42.6
LightRetriever-Qwen2.5-1.5B-Dense (Ma et al., 2026)	233.0M	51.2	38.3	36.1	33.0	82.1	41.2	59.0	39.1
RAVF-Qwen2.5-1.5B-0.9-Dense	54.6M	39.7	33.2	27.1	29.0	73.1	38.4	51.7	36.1
RAVF-Qwen2.5-1.5B-0.5-Dense	32.3M	38.5	32.7	26.9	28.1	70.5	38.1	49.1	35.1
RAVF-Qwen2.5-1.5B-0.1-Dense	9.2M	32.8	27.8	23.7	15.9	53.1	32.7	32.2	24.0
Qwen2.5-3B									
Full-Qwen2.5-3B	3B	55.5	47.0	44.1	45.8	89.6	52.9	71.6	43.7
LightRetriever-Qwen2.5-3B-Dense (Ma et al., 2026)	310.6M	49.8	39.6	41.1	35.0	83.9	45.4	62.8	41.0
RAVF-Qwen2.5-3B-0.9-Dense	72.8M	48.7	36.4	29.4	30.8	76.4	41.9	58.1	39.1
RAVF-Qwen2.5-3B-0.5-Dense	44.7M	47.9	35.8	29.1	29.3	74.7	41.4	56.0	38.0
RAVF-Qwen2.5-3B-0.1-Dense	12.3M	43.2	31.1	25.0	16.8	58.2	35.6	39.8	27.8
Qwen2.5-7B									
Full-Qwen2.5-7B	7B	57.1	48.2	43.5	47.9	90.7	56.5	74.7	44.8
LightRetriever-Qwen2.5-7B-Dense (Ma et al., 2026)	543.6M	53.3	40.0	39.1	36.8	85.1	48.8	64.4	41.8
RAVF-Qwen2.5-7B-0.9-Dense	127.4M	50.5	36.8	33.2	32.7	77.8	45.1	59.3	39.4
RAVF-Qwen2.5-7B-0.5-Dense	75.5M	49.9	36.3	32.4	31.9	75.7	44.5	57.2	38.4
RAVF-Qwen2.5-7B-0.1-Dense	21.5M	44.7	31.4	28.9	20.4	62.4	39.7	43.0	28.5

Table 14: BEIR detailed results, part II. Detailed per-task BEIR retrieval results. All entries are nDCG@10. Datasets: NFC=NFCorpus, SCID=SCIDOCS, SciF=SciFact, TREC=TRECCOVID, T20=Touche2020.

Method	Params	NFC	NQ	Quora	SCID	SciF	TREC	T20	Avg.
Static Embedding (Aarsen, 2025)	31.3M	30.0	23.1	77.5	13.2	59.3	44.6	22.4	34.1
BM25 (Robertson et al., 1994)	NN Free	32.5	32.9	78.9	15.8	66.5	65.6	36.7	41.7
USTAD (Kim et al., 2024)	66M	30.7	50.8	81.4	14.4	55.5	72.3	24.7	44.2
BGE-M3-Dense (Chen et al., 2024)	568M	31.5	60.1	88.4	15.2	63.6	47.0	20.6	47.2
BGE-M3-Sparse (Chen et al., 2024)	568M	28.3	20.3	73.4	12.2	63.3	52.7	27.8	39.7
SPLADE-v3 (Lassance et al., 2024)	435M	35.7	58.6	81.4	15.8	71.0	74.8	29.3	50.2
LLM2Vec-Llama8B (BehnamGhader et al., 2024)	8B	41.8	64.2	87.2	23.0	78.2	80.3	20.5	56.6
E5-Mistral7B (Wang et al., 2024)	7B	38.6	63.5	89.6	16.3	76.4	87.2	26.4	56.9
Llama3.2-1B									
Full-Llama3.2-1B	1B	35.5	58.6	89.4	19.8	72.3	66.7	20.9	53.1
LightRetriever-Llama3.2-1B-Dense (Ma et al., 2026)	262.7M	30.1	52.2	87.0	17.2	66.5	68.8	24.7	48.7
RAVF-Llama3.2-1B-0.9-Dense	65.6M	27.0	51.6	85.5	16.2	65.4	69.7	25.5	45.9
RAVF-Llama3.2-1B-0.5-Dense	41.4M	26.7	50.6	84.8	15.8	64.1	69.8	24.4	45.0
RAVF-Llama3.2-1B-0.1-Dense	12.8M	20.9	40.2	78.4	12.9	57.1	59.6	22.1	37.4
Llama3.2-3B									
Full-Llama3.2-3B	3B	37.6	62.8	89.5	22.5	75.3	68.7	21.2	55.6
LightRetriever-Llama3.2-3B-Dense (Ma et al., 2026)	394.0M	31.7	56.4	87.8	19.0	70.4	68.6	21.6	51.0
RAVF-Llama3.2-3B-0.9-Dense	98.3M	28.8	55.0	86.6	17.4	65.9	71.2	23.8	48.2
RAVF-Llama3.2-3B-0.5-Dense	62.1M	28.2	54.3	86.1	17.0	65.7	71.1	24.2	47.6
RAVF-Llama3.2-3B-0.1-Dense	19.1M	23.3	45.4	81.1	14.5	59.7	60.5	19.8	40.4
Llama3.1-8B									
Full-Llama3.1-8B	8B	38.6	65.5	89.9	23.8	77.3	62.0	22.2	56.8
LightRetriever-Llama3.1-8B-Dense (Ma et al., 2026)	525.3M	31.8	59.0	88.5	19.3	70.6	65.7	21.1	52.0
LightRetriever-Llama3.1-8B-Hybrid (Ma et al., 2026)	525.3M	34.6	61.9	88.6	20.7	72.6	71.5	27.8	54.4
RAVF-Llama3.1-8B-0.9-Dense	131.3M	29.3	57.6	86.1	17.4	68.2	73.0	22.4	50.4
RAVF-Llama3.1-8B-0.5-Dense	83.1M	28.6	56.8	85.7	16.8	67.2	72.5	22.4	49.7
RAVF-Llama3.1-8B-0.1-Dense	25.6M	23.2	48.6	80.8	14.5	61.8	60.1	20.2	42.6
RAVF-Llama8B-0.9-Hybrid	131.3M	32.1	60.6	87.3	18.8	71.4	73.2	25.7	53.1
RAVF-Llama8B-0.5-Hybrid	83.1M	31.8	60.2	86.9	18.5	70.8	72.8	25.4	52.7
RAVF-Llama8B-0.1-Hybrid	25.6M	27.9	57.0	83.7	17.1	67.4	65.2	22.8	49.5
Qwen2.5-1.5B									
Full-Qwen2.5-1.5B	1.5B	36.4	57.1	89.0	20.4	72.7	73.0	28.4	54.0
LightRetriever-Qwen2.5-1.5B-Dense (Ma et al., 2026)	233.0M	29.8	50.9	86.7	18.1	66.5	72.6	29.1	48.9
RAVF-Qwen2.5-1.5B-0.9-Dense	54.6M	26.5	49.5	84.6	15.9	63.7	68.2	28.5	44.3
RAVF-Qwen2.5-1.5B-0.5-Dense	32.3M	26.0	48.5	84.0	15.6	62.4	67.3	27.8	43.4
RAVF-Qwen2.5-1.5B-0.1-Dense	9.2M	20.5	37.8	77.1	12.6	55.4	56.6	23.8	35.1
Qwen2.5-3B									
Full-Qwen2.5-3B	3B	35.9	61.4	89.5	22.3	72.9	65.7	26.0	54.9
LightRetriever-Qwen2.5-3B-Dense (Ma et al., 2026)	310.6M	30.6	54.5	87.5	18.5	64.8	71.2	26.1	50.1
RAVF-Qwen2.5-3B-0.9-Dense	72.8M	27.4	53.3	85.9	16.8	62.3	66.3	24.6	46.5
RAVF-Qwen2.5-3B-0.5-Dense	44.7M	27.0	52.8	85.3	16.5	61.1	64.9	25.0	45.7
RAVF-Qwen2.5-3B-0.1-Dense	12.3M	21.9	42.5	78.8	13.4	55.2	55.8	23.8	37.9
Qwen2.5-7B									
Full-Qwen2.5-7B	7B	37.2	63.7	89.7	23.7	77.2	69.1	25.3	56.6
LightRetriever-Qwen2.5-7B-Dense (Ma et al., 2026)	543.6M	30.7	55.9	87.6	19.5	67.9	64.1	24.1	50.6
RAVF-Qwen2.5-7B-0.9-Dense	127.4M	27.6	54.4	86.0	18.4	64.9	61.8	22.8	47.4
RAVF-Qwen2.5-7B-0.5-Dense	75.5M	27.4	53.5	85.5	18.0	64.3	61.4	22.6	46.6
RAVF-Qwen2.5-7B-0.1-Dense	21.5M	21.6	43.7	79.8	15.0	58.5	52.6	21.3	39.4

Table 15: CMTEB-R detailed results. Detailed per-task retrieval results. All entries are nDCG@10.

Method	Params	Cmedqa	Covid	DuReader	Ecom	MMarco	Medical	T2	Video	Avg.
Static Embedding (Aarsen, 2025)	31.3M	7.5	45.7	37.9	33.9	34.9	15.7	35.8	38.9	31.3
BM25 (Robertson et al., 1994)	NN Free	13.7	86.6	57.1	45.1	48.3	32.1	60.5	62.7	50.8
BGE-M3-Dense (Chen et al., 2024)	568M	31.0	77.2	82.9	57.5	76.9	51.5	80.8	54.4	64.0
BGE-M3-Sparse (Chen et al., 2024)	568M	24.5	76.0	71.4	50.3	59.2	44.0	71.7	58.5	57.0
LLM2Vec-Llama8B (BehnamGhader et al., 2024)	8B	35.2	16.5	80.7	54.4	76.5	54.6	65.2	52.4	54.4
E5-Mistral7B (Wang et al., 2024)	7B	34.2	73.1	87.0	46.0	74.8	52.8	80.7	45.4	61.8
Llama3.2-1B										
Full-Llama3.2-1B	1B	32.3	74.2	84.6	53.8	73.5	50.5	78.0	63.1	63.8
LightRetriever-Llama3.2-1B-Dense (Ma et al., 2026)	262.7M	27.2	67.6	80.8	51.0	67.9	43.0	73.4	58.5	58.7
RAVF-Llama3.2-1B-0.9-Dense	65.6M	27.5	65.2	79.6	50.3	66.3	42.8	72.4	57.2	57.6
RAVF-Llama3.2-1B-0.5-Dense	41.4M	27.3	62.0	78.4	47.4	64.9	41.8	70.2	53.8	55.7
RAVF-Llama3.2-1B-0.1-Dense	12.M	22.5	55.5	69.3	33.5	53.8	34.3	57.1	37.4	45.4
Llama3.2-3B										
Full-Llama3.2-3B	3B	36.8	75.6	87.9	57.7	75.7	53.8	81.1	60.5	66.1
LightRetriever-Llama3.2-3B-Dense (Ma et al., 2026)	394.0M	29.8	70.2	84.0	51.7	69.2	46.5	75.7	56.4	60.4
RAVF-Llama3.2-3B-0.9-Dense	98.3M	29.7	67.9	82.6	50.1	67.6	46.4	75.1	55.2	59.3
RAVF-Llama3.2-3B-0.5-Dense	62.1M	29.5	66.7	81.8	48.1	66.5	45.9	73.6	51.7	58.0
RAVF-Llama3.2-3B-0.1-Dense	19.1M	25.7	61.0	75.0	35.9	56.1	38.5	63.7	37.3	49.2
Llama3.1-8B										
Full-Llama3.1-8B	8B	39.5	77.1	89.5	58.9	78.1	57.6	84.1	56.2	67.6
LightRetriever-Llama3.1-8B-Dense (Ma et al., 2026)	525.3M	32.1	71.0	85.7	53.0	70.2	49.0	78.5	52.0	61.4
LightRetriever-Llama3.1-8B-Hybrid (Ma et al., 2026)	525.3M	32.1	72.8	86.6	55.3	71.6	49.7	80.1	55.8	63.0
RAVF-Llama3.1-8B-0.9-Dense	131.1M	31.9	69.6	84.4	52.0	68.6	48.7	77.5	52.0	60.6
RAVF-Llama3.1-8B-0.5-Dense	83.1M	31.8	67.9	83.5	49.4	67.3	47.4	76.0	49.3	59.1
RAVF-Llama3.1-8B-0.1-Dense	25.6M	27.8	61.7	77.6	37.3	57.5	40.5	66.0	34.6	50.4
RAVF-Llama8B-0.9-Hybrid	131.1M	31.2	71.7	85.9	54.2	70.6	49.6	79.4	54.9	62.2
RAVF-Llama8B-0.5-Hybrid	83.1M	31.1	70.7	85.3	52.7	69.8	48.8	78.5	53.0	61.2
RAVF-Llama8B-0.1-Hybrid	25.6M	29.2	67.8	82.4	46.3	65.2	45.1	73.4	43.7	56.6
Qwen2.5-1.5B										
Full-Qwen2.5-1.5B	1.5B	37.9	78.2	87.6	56.9	75.9	54.7	78.8	57.0	65.9
LightRetriever-Qwen2.5-1.5B-Dense (Ma et al., 2026)	233.0M	31.3	73.8	85.3	52.8	71.8	49.5	75.2	55.7	61.9
RAVF-Qwen2.5-1.5B-0.9-Dense	54.6M	27.2	47.6	78.1	43.1	58.7	46.3	64.1	42.2	50.9
RAVF-Qwen2.5-1.5B-0.5-Dense	32.3M	24.3	42.1	74.7	36.1	54.6	42.8	56.8	33.5	45.6
RAVF-Qwen2.5-1.5B-0.1-Dense	9.2M	13.7	35.5	62.9	22.0	40.9	28.4	39.1	19.7	32.8
Qwen2.5-3B										
Full-Qwen2.5-3B	3B	39.4	80.1	89.5	61.9	77.8	58.7	84.4	63.6	69.4
LightRetriever-Qwen2.5-3B-Dense (Ma et al., 2026)	310.6M	32.7	76.0	86.8	56.7	73.2	51.6	80.9	59.5	64.7
RAVF-Qwen2.5-3B-0.9-Dense	72.8M	31.2	50.1	80.3	48.0	61.4	48.0	70.3	50.0	54.9
RAVF-Qwen2.5-3B-0.5-Dense	44.7M	29.9	43.8	78.0	45.0	58.6	46.4	66.1	44.1	51.5
RAVF-Qwen2.5-3B-0.1-Dense	12.3M	20.6	34.7	66.4	28.9	44.7	34.0	50.4	27.5	38.4
Qwen2.5-7B										
Full-Qwen2.5-7B	7B	41.3	81.4	91.0	61.1	80.0	59.8	85.5	60.3	70.1
LightRetriever-Qwen2.5-7B-Dense (Ma et al., 2026)	543.6M	34.3	74.6	88.0	55.7	75.4	53.2	81.6	57.2	65.0
RAVF-Qwen2.5-7B-0.9-Dense	127.4M	31.8	48.5	81.3	46.8	62.4	50.5	70.8	46.4	54.8
RAVF-Qwen2.5-7B-0.5-Dense	75.5M	30.1	43.8	78.5	43.0	59.7	47.5	66.2	40.7	51.2
RAVF-Qwen2.5-7B-0.1-Dense	21.5M	20.6	37.5	67.7	29.9	46.9	37.5	51.4	24.7	39.5

Table 16: Per-task core ablation on the BEIR-6 with Llama3.1-8B. The appendix uses the same task set and averages as Table 3 in the main paper.

Retain	Method	TRECCOVID		FiQA		SCIDOCS		ArguAna		SciFact		NFCorpus		Avg.	
		nDCG	R@100	nDCG	R@100	nDCG	R@100	nDCG	R@100	nDCG	R@100	nDCG	R@100	nDCG	R@100
Orig.	lightretriever Dense / FullBag	0.6570	–	0.4980	–	0.1930	–	0.5580	–	0.7060	–	0.3180	–	0.4883	–
0.9	Random deletion	0.6827	0.1228	0.4405	0.7414	0.1595	0.3840	0.5256	0.9716	0.6611	0.9120	0.2719	0.2585	0.4569	0.5650
0.9	Frequency deletion	0.7335	0.1331	0.4772	0.7767	0.1700	0.3962	0.5330	0.9772	0.6769	0.9137	0.2917	0.2740	0.4804	0.5785
0.9	Utility deletion	0.7347	0.1332	0.4708	0.7707	0.1704	0.3956	0.5332	0.9765	0.6772	0.9137	0.2900	0.2727	0.4794	0.5771
0.9	Utility + Folding	0.7358	0.1329	0.4844	0.7795	0.1724	0.4007	0.5357	0.9787	0.6831	0.9237	0.2937	0.2735	0.4842	0.5815
0.9	Utility + Folding + Protection	0.7338	0.1326	0.4844	0.7795	0.1724	0.4007	0.5357	0.9787	0.6818	0.9237	0.2937	0.2735	0.4836	0.5815
0.9	Utility + Folding + Protection + InfoNCE	0.7296	0.1361	0.4860	0.7783	0.1739	0.4048	0.5537	0.9794	0.6816	0.9237	0.2930	0.2746	0.4863	0.5828
0.5	Random deletion	0.3508	0.0805	0.2254	0.5015	0.1018	0.2689	0.4076	0.9019	0.4953	0.8147	0.1779	0.2076	0.2931	0.4625
0.5	Frequency deletion	0.7037	0.1299	0.4279	0.7463	0.1513	0.3644	0.5178	0.9666	0.6391	0.9043	0.2514	0.2601	0.4485	0.5619
0.5	Utility deletion	0.7116	0.1302	0.4183	0.7418	0.1505	0.3673	0.5203	0.9673	0.6389	0.9037	0.2574	0.2546	0.4495	0.5608
0.5	Utility + Folding	0.7305	0.1332	0.4760	0.7715	0.1651	0.3965	0.5289	0.9737	0.6730	0.9197	0.2866	0.2746	0.4767	0.5782
0.5	Utility + Folding + Protection	0.7306	0.1326	0.4798	0.7734	0.1658	0.3963	0.5294	0.9751	0.6742	0.9230	0.2869	0.2727	0.4778	0.5788
0.5	Utility + Folding + Protection + InfoNCE	0.7248	0.1353	0.4802	0.7732	0.1677	0.4000	0.5508	0.9787	0.6719	0.9230	0.2856	0.2726	0.4802	0.5805
0.1	Random deletion	0.0348	0.0134	0.0219	0.0917	0.0169	0.0699	0.0950	0.4026	0.0932	0.3343	0.0423	0.0845	0.0507	0.1661
0.1	Frequency deletion	0.3269	0.0604	0.2242	0.4970	0.0745	0.2233	0.3869	0.8862	0.3629	0.6979	0.1004	0.1681	0.2460	0.4222
0.1	Utility deletion	0.3568	0.0625	0.2231	0.4968	0.0781	0.2324	0.3960	0.8940	0.3595	0.7234	0.1107	0.1678	0.2540	0.4295
0.1	Utility + Folding	0.5842	0.1027	0.4021	0.7140	0.1298	0.3329	0.4716	0.9445	0.5630	0.8537	0.2145	0.2390	0.3942	0.5311
0.1	Utility + Folding + Protection	0.5987	0.1085	0.4298	0.7396	0.1437	0.3530	0.4889	0.9509	0.6151	0.8730	0.2324	0.2400	0.4181	0.5442
0.1	Utility + Folding + Protection + InfoNCE	0.6013	0.1123	0.4382	0.7325	0.1452	0.3566	0.5237	0.9694	0.6177	0.8777	0.2325	0.2419	0.4264	0.5484

Table 17: Per-task pruning method comparison on the BEIR-6 subset with Llama3.1-8B. nDCG@10 and Recall@100 are reported next to each other for each task.

Retain	Method	TRECCOVID		FiQA		SCIDOCS		ArguAna		SciFact		NFCorpus		Avg.	
		nDCG	R@100	nDCG	R@100	nDCG	R@100	nDCG	R@100	nDCG	R@100	nDCG	R@100	nDCG	R@100
Orig.	lightretriever Dense / FullBag	0.6570	–	0.4980	–	0.1930	–	0.5580	–	0.7060	–	0.3180	–	0.4883	–
0.9	Random deletion	0.6827	0.1228	0.4405	0.7414	0.1595	0.3840	0.5256	0.9716	0.6611	0.9120	0.2719	0.2586	0.4569	0.5651
0.9	Frequency deletion	0.7335	0.1331	0.4772	0.7767	0.1700	0.3962	0.5330	0.9772	0.6769	0.9137	0.2917	0.2740	0.4804	0.5785
0.9	IDF deletion	0.4584	0.0900	0.2465	0.5157	0.1208	0.3004	0.4138	0.9146	0.5569	0.8657	0.2409	0.2344	0.3396	0.4868
0.9	Magnitude deletion	0.6822	0.1205	0.4299	0.7401	0.1651	0.3953	0.4989	0.9744	0.6552	0.8910	0.2695	0.2583	0.4501	0.5633
0.9	Taylor deletion	0.7200	0.1320	0.4859	0.7803	0.1715	0.3987	0.5351	0.9787	0.6808	0.9203	0.2930	0.2735	0.4811	0.5806
0.9	RAVF (without InfoNCE)	0.7338	0.1326	0.4844	0.7795	0.1724	0.4007	0.5357	0.9787	0.6818	0.9237	0.2937	0.2735	0.4836	0.5815
0.5	Random deletion	0.3508	0.0805	0.2254	0.5015	0.1018	0.2689	0.4076	0.9019	0.4953	0.8147	0.1779	0.2077	0.2931	0.4625
0.5	Frequency deletion	0.7037	0.1299	0.4279	0.7463	0.1513	0.3644	0.5178	0.9666	0.6391	0.9043	0.2514	0.2601	0.4485	0.5619
0.5	IDF deletion	0.0486	0.0089	0.0479	0.1500	0.0339	0.1075	0.1761	0.6230	0.1897	0.4597	0.0738	0.0955	0.0950	0.2408
0.5	Magnitude deletion	0.3197	0.0624	0.1740	0.4228	0.1052	0.2937	0.3975	0.9132	0.4251	0.7236	0.1446	0.1805	0.2610	0.4327
0.5	Taylor deletion	0.6065	0.1226	0.4327	0.7446	0.1404	0.3496	0.5056	0.9666	0.6140	0.8780	0.2411	0.2434	0.4234	0.5508
0.5	RAVF (without InfoNCE)	0.7306	0.1326	0.4798	0.7734	0.1658	0.3963	0.5294	0.9751	0.6742	0.9230	0.2869	0.2727	0.4778	0.5788
0.1	Random deletion	0.0348	0.0134	0.0215	0.0902	0.0167	0.0701	0.0950	0.4026	0.0932	0.3309	0.0415	0.0851	0.0504	0.1654
0.1	Frequency deletion	0.3269	0.0604	0.2242	0.4970	0.0745	0.2233	0.3869	0.8862	0.3629	0.6979	0.1004	0.1666	0.2460	0.4219
0.1	IDF deletion	0.0038	0.0012	0.0049	0.0162	0.0037	0.0182	0.0288	0.1437	0.0243	0.0842	0.0145	0.0422	0.0133	0.0509
0.1	Magnitude deletion	0.0116	0.0045	0.0152	0.0641	0.0152	0.0655	0.0736	0.3535	0.0809	0.2953	0.0210	0.0433	0.0363	0.1377
0.1	Taylor deletion	0.1959	0.0393	0.1716	0.4093	0.0556	0.1544	0.2295	0.6743	0.3361	0.6602	0.1278	0.1571	0.1861	0.3491
0.1	RAVF (without InfoNCE)	0.5987	0.1085	0.4298	0.7396	0.1437	0.3530	0.4889	0.9509	0.6151	0.8730	0.2324	0.2400	0.4181	0.5442