Query-Aware Index Pruning for Retrieval Under Budget Constraints

Stefano Campese
University of Trento
Trento, Italy
stefano.campese@unitn.it

Ivano Lauriola
Amazon AGI
El Segundo, USA
lauivano@amazon.com

Alessandro Moschitti
Amazon AGI
El Segundo, USA
amosch@amazon.com

Abstract

Production search indices operate under strict storage and latency budgets that prevent indexing the full corpus (e.g.: web documents). When only a fraction of documents can be retained, the system must decide which ones to keep. Existing pruning methods make this decision using query-agnostic signals, including lexical quality, embedding magnitude, and corpus centrality or single-feature query-aware heuristics (e.g., query similarity, nearest-neighbour coverage), without optimising which documents actually contribute to answering the queries the system will serve. We recast index pruning as query-aware subset selection: given a fixed budget and an expected query workload, the goal is to retain the document subset that maximises retrieval performance for the queries the system will serve. We introduce Query-aware Document Selection (QDS), a lightweight framework that scores documents using 13 retrieval-structural features derived from query–document interactions and index redundancy, capturing how many queries a document serves, whether viable alternatives exist, and how replaceable the document is within the retained set. A linear scorer trained with Evolution Strategies optimises nDCG@10 of the pruned index end-to-end, requiring no differentiable surrogate while remaining fully interpretable. Across 10 IR benchmarks (1K–120M documents), four dense encoders plus BM25, and three modalities (text, image, audio), QDS consistently outperforms query-agnostic and single-feature query-aware baselines under aggressive pruning budgets. We show that coverage-oriented heuristics degrade at web scale by over-selecting high-frequency hard negatives, i.e., documents that are nearest neighbours to many queries but relevant to none. Given its simplicity, QDS can operate on existing indexes of 100M-1B documents in minutes with limited hardware.

CCS Concepts

• **Information systems → Retrieval models and ranking**; **Search index compression**; *Search engine indexing*.

Keywords

Index Pruning, Dense Retrieval, Query-Aware Selection, Evolution Strategies, Efficient Retrieval

CIKM '26, Rome, Italy

ACM ISBN 979-8-4007-2539-5/2026/11
https://doi.org/10.1145/3799682.3840963

ACM Reference Format:
Stefano Campese, Ivano Lauriola, and Alessandro Moschitti. 2026. Query-Aware Index Pruning for Retrieval Under Budget Constraints. In *Proceedings of the 35th ACM International Conference on Information and Knowledge Management (CIKM '26), November 07–11, 2026, Rome, Italy*. ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3799682.3840963

1 Introduction

Dense retrieval encodes documents and queries into a shared embedding space for nearest-neighbour lookup [22, 41], and has become the dominant first stage in both classical IR and retrieval-augmented generation pipelines.[1] As corpora grow to millions or billions of passages, every production system faces a hard budget: GPU memory, serving latency, or infrastructure cost caps the number of documents that can be indexed. Reducing the index while preserving retrieval quality, *index pruning*, is therefore a core operational problem at the intersection of information access and resource-efficient AI. At web scale, this constraint is unavoidable: modern retrieval systems operate over corpora of billions of documents that far exceed what can be fully embedded and served under practical hardware budgets.

The natural response has been to score documents by query-agnostic properties: lexical quality, such as Information-to-Noise ratio [45] (ITN), embedding magnitude [19], supervised passage-quality estimators [11], and geometric heuristics such as centroid distance or k-NN density. All of these treat every document as equally likely to be needed. But production query traffic is heavily skewed: a small fraction of query types drives most of the volume, and only a subset of documents is critical to answering them. Under a fixed budget, the question is therefore not *which documents are intrinsically good?* but *which documents serve the queries users will actually issue?* When this mismatch is ignored, the consequences can be severe: on MSMARCO v1 (8.8M passages), coverage-based selection, a widely used query-aware heuristic, plateaus at 14–17% retention, *worse than uniform random sampling*, because it fills the index with high-frequency hard negatives rather than documents that answer real queries.

We introduce QDS (Query-aware Document Selection). The core idea is to score each document by its marginal contribution to retrieval quality under the expected query workload, rather than by intrinsic properties. We operationalise this through 13 per-document features that directly measure workload dependency: how many queries a document serves as top result, whether backup documents exist for those queries, how redundant the document is with the rest of the index, and how similar it is to documents known to be relevant. These features are combined by a linear scorer with just 13 learnable weights, trained via Evolution Strategies [31] to

[1] Code and data: https://github.com/sirCamp/qds-index-pruning

maximise the nDCG@10 of the resulting pruned index under a fixed budget B (B-document index). Unlike prior work, which either applies heuristics without optimisation (Coverage, Magnitude) or trains a classifier on a proxy objective ("is this passage relevant to some query?" [11]), QDS optimises the actual retrieval metric of the pruned index end-to-end: the reward is the nDCG@10 of running retrieval on the selected subset, with no differentiable surrogate in the loop. Because the scorer has only 13 weights, each one is directly readable: a positive weight means "prefer documents with this property," a negative weight means "avoid them." This lets practitioners inspect the selection strategy per dataset. For instance, on adversarial retrieval tasks the optimiser assigns a *negative* weight to coverage, learning that popular nearest neighbours are distractors to be removed, not documents to be retained. QDS is orthogonal to per-document compression (PQ [21], Matryoshka [23], dimension pruning [34]) and to ANN structures (e.g.: HNSW [44]), which compose multiplicatively in a deployed system; it also serves as an analysis instrument for studying query-aware pruning across retrievers, scales, and modalities.

Our contribution is QDS, a query-aware document selection framework that combines 13 inspectable retrieval-structural features through a 13-parameter linear scorer trained with Evolution Strategies to directly maximise the non-differentiable nDCG@10 reward of the pruned index. We validate QDS along four axes. *Scale and modality*: across 10 corpora spanning five orders of magnitude (1K–120M documents), 4 dense encoders, BM25, and two non-text modalities, QDS outperforms every query-agnostic and single-feature query-aware baseline at aggressive pruning budgets, and retains 85.7% of full-corpus performance at 120M passages with only 30% of documents indexed (Sections 4.2–4.6). *Comparison to neural quality estimation*: a 13-parameter linear scorer beats a retrained 220M-parameter T5 classifier on 9 of 10 dataset–budget cells with 5–15× lower variance (Section 5.1). *Decomposition*: the feature framework alone contributes +5–15 pp over single-feature heuristics, with ES adding a further +5–7 pp over evaluation-matched random search (Section 5.2). *Interpretability*: per-dataset weights expose the learned strategy directly, revealing for instance that QDS assigns a negative weight to coverage on ArguAna's adversarial structure, driving the >100% Less-is-More retention regime (Section 5.3).

2 Related Work

2.1 Index Pruning

Reducing index size has a long history in information retrieval. Classic *static index pruning* for inverted indices removes posting list entries [9] or entire documents [7] based on term-level statistics, and query logs have long been used to guide those decisions: Lam et al. [24] showed that query-log information substantially improves over query-agnostic pruning, Garcia [18] prunes posting lists by how often a document appears in the top-k results of past queries, and Altingovde et al. [2] combine the same document-popularity counts with query views to prune entire documents. Our `coverage_count` feature (f_9), and the Coverage baseline built on it, are the dense-retrieval analogue of that signal. We differ in two respects: we operate on embedding geometry rather than term statistics, and we show that the count alone is insufficient and can even be harmful, requiring combination with redundancy, margin, and fallback signals through a learned, task-dependent scorer.

With the rise of dense retrieval [22, 41], Chang et al. [11] proposed *neural passage quality estimation*: a T5 model trained on MSMARCO predicts whether a passage is relevant to *any* query, enabling query-agnostic pruning across pipelines. Their score is an absolute notion of passage quality, whereas we learn a *relative* function ranking documents by utility to a specific workload. We use their largest released checkpoint, `qt5-base` (~220M parameters), as a *zero-shot* baseline (**QualT5-zs**), and for a head-to-head comparison re-train T5-base on two BEIR collections, ArguAna and SciDocs, chosen for their adversarial-similarity and citation-network structure. This *approximates* Chang et al.'s setup, substituting BEIR qrels for MSMARCO-style triples, which BEIR does not provide, while keeping their prompt and language-model head reduction (Section 5.1).

Separately, Hofstätter et al. [19] observed that bi-encoders trained without embedding normalisation (TAS-B) encode a quality signal in the L2 norm of the document vector, which Chang et al. [11] reuse as a pruning baseline. We include this as Magnitude, alongside ITN (unique-to-total term ratio) [45] and two geometric heuristics (centroid distance, k-NN density) that appear as informal baselines in prior work without a single attribution.

Several recent lines prune at granularities orthogonal to document selection: Matryoshka representations [23] and product quantisation [21] compress per-document storage; Siciliano et al. [34] and Faggioli et al. [15] prune embedding *dimensions* (offline and per-query); Acquavia et al. [1] prune *token-level* ColBERT embeddings; Tonellotto and Macdonald [36] trim the candidate pool at query time. None reduces the number of indexed documents, and all compose with document-level selection: a deployed system can apply QDS to choose *which* documents to index, then any of the above to shrink *how* they are stored.

2.2 Document-Level Signals and Optimisation

Our features draw on several traditions. *Retrievability* [3] quantifies how easily a document can be found, recently extended with topic-focused measures [10]. *Coverage-based selection*, inspired by set cover [16], retains documents serving as nearest neighbours to the most queries; we use it as one of 13 features rather than the sole criterion, and show that the single-criterion version is actively harmful when nearest neighbours are distractors. *Redundancy and diversity*, studied through MMR [8] and result diversification [32], traditionally operate at query time; our features adapt them to corpus construction, addressing which documents to store rather than which results to show.

We optimise the linear scorer with Evolution Strategies (ES) [31], a derivative-free method that estimates gradients of an expected reward through Gaussian perturbations. ES fits because the reward requires running the full retrieval pipeline on a discrete document subset: both the top-B selection and nDCG@10 are non-differentiable, leaving no path from parameters to metric. Prior approaches to non-differentiable ranking metrics do not transfer. LambdaRank and its descendants [6] define lambda gradients over document pairs within a fixed candidate list, but this presupposes a stable candidate set per query, whereas here changing $\mathbf{w}$ changes

the candidate pool itself. Closer to our setting, ES-Rank [20] learns a ranking function against IR metrics with evolution strategies, but optimises a ranker over a fixed corpus rather than the corpus itself. The same structure drives a renewed wave of evolutionary methods in reinforcement learning: robust gradients for noisy rewards [12], hard-thresholded ES for sparse policies [17], proximal CMA-ES [28], a survey of ∼100 studies [27], and REINFORCE-style preference optimisation [33]. Our contribution here is application-oriented rather than algorithmic: a 13-parameter linear scorer rather than a deep policy, with the same Gaussian-perturbation / scalar-reward loop applied, to our knowledge for the first time, to budget-constrained index pruning.

3 Method

3.1 Problem Formulation

Consider a dense retrieval system with a corpus $C = \{d_1, \ldots, d_N\}$ of N documents, an embedding model ϕ that maps both documents and queries to $\mathbb{R}^m$, and a set of training queries $Q = \{q_1, \ldots, q_M\}$ with associated relevance judgments. Given a *budget* $B < N$ (e.g., $B = 0.1N$ for a 10% budget), we seek the subset $\mathcal{S}^* \subseteq C$ of exactly B documents that maximizes retrieval quality when used as the dense index:

$$\mathcal{S}^* = \arg\max_{\substack{\mathcal{S}\subseteq C \\ |\mathcal{S}|=B}} \frac{1}{|Q|} \sum_{q\in Q} \text{nDCG@10}(q,\ \text{Retrieve}(\phi(q), \mathcal{S})) \quad (1)$$

where $\text{Retrieve}(\phi(q), \mathcal{S})$ returns the top-10 documents from $\mathcal{S}$ ranked by cosine similarity to $\phi(q)$.

Linear relaxation. We relax this discrete optimisation into a scoring problem. Each document receives a scalar score $s(d) = \mathbf{w}^\top \mathbf{f}(d)$, where $\mathbf{f}(d) \in \mathbb{R}^{13}$ is a feature vector (Section 3.2) and $\mathbf{w} \in \mathbb{R}^{13}$ are learned weights, yielding a deterministic selection:

$$\mathcal{S}(\mathbf{w}) = \text{top-}B\Big(\{s(d_i)\}_{i=1}^N\Big) \quad (2)$$

We adopt this linear form for its 13-parameter interpretability and inductive bias against overfitting; non-linear variants (MLP head) are evaluated in Section 5.2. QDS trains one weight vector $\mathbf{w}$ per budget B independently, since the optimal trade-off between coverage, redundancy, and relevance shifts with how aggressive the pruning is.

3.2 Document Features

We characterise each document through 13 features spanning six complementary axes, query utility, retrieval frequency, corpus redundancy, coverage of unique intents, diversity, and relevance anchoring, all computed once from the training query set Q and corpus embeddings using GPU-accelerated batched operations. Let $\text{sim}(q, d) = \cos(\phi(q), \phi(d))$ denote cosine similarity in the embedding space, $\text{NN}(q)$ the nearest corpus document for query q, and $\text{kNN}(d)$ the k=10 nearest documents to d. Table 1 summarises all 13 features; the families are described below.

Query Relevance (f_1–f_3). The most direct selection signal is how relevant a document is to the queries the system serves [22]. f_1 captures the peak similarity to any training query (a document with low f_1 is far from every known information need and is a strong removal candidate); f_2 measures broad-coverage relevance (documents above the mean across queries rather than peak-relevant to one); f_3 with k=10 interpolates between the two and surfaces documents that serve a focused cluster of related queries, a pattern common in specialised domains like finance (FiQA) or biomedicine (NFCorpus).

Table 1: The 13 QDS features. All are min-max normalised to $[0, 1]$ across the corpus, so learned weights are comparable in scale.

ID	Name	Definition (sketch)	Family
f_1	max_query_sim	$\max_{q\in Q} \text{sim}(q, d)$	Query Relevance
f_2	mean_query_sim	$\overline{\text{sim}(q, d)}$ over $q \in Q$	Query Relevance
f_3	top_k_query_sim	Mean of top-10 query similarities	Query Relevance
f_4	retrievability	$\lvert\{q : d \in \text{top-100}(q)\}\rvert$	Retrievability
f_5	weighted_ret	Rank-discounted retrievability (nDCG-style)	Retrievability
f_6	redundancy_mean	Mean sim to $\text{kNN}(d)$	Redundancy
f_7	redundancy_max	Max sim to any other corpus doc	Redundancy
f_8	isolation	$1 - f_7$ (complement)	Redundancy
f_9	coverage_count	$\lvert\{q : d = \text{NN}(q)\}\rvert$	Coverage
f_{10}	coverage_margin	Mean gap to 2nd-NN over queries d tops	Coverage
f_{11}	backup_count	$\lvert\{q : d \in \text{top-3}(q) \setminus \{\text{NN}(q)\}\}\rvert$	Coverage
f_{12}	uniqueness	$1 - f_6$ (complement)	Diversity
f_{13}	relevant_doc_sim	$\phi(d)^\top \bar{\phi}_{rel} / \lVert\bar{\phi}_{rel}\rVert$ over qrels	Anchoring

Retrievability (f_4, f_5). Adapted from Azzopardi and Vinay [3]: a document may be relevant yet effectively invisible if other documents consistently outrank it. f_4 counts the training queries for which d appears among the top-100 nearest neighbours; f_5 rank-discounts the count so high-rank appearances dominate, analogous to the gain function in nDCG:

$$f_5(d) = \sum_{q\in Q} \frac{\mathbb{1}[d \in \text{top-100}(q)]}{\log_2(\text{rank}(d, q) + 1)}. \quad (3)$$

Redundancy (f_6–f_8). Under budget constraints, retaining two near-identical documents wastes a slot. These features adapt the intuition behind MMR [8] and result diversification [32] from query-time ranking to corpus-time index construction. f_6 averages similarity to the 10 nearest corpus neighbours (high values indicate dense regions where documents are interchangeable); f_7 measures similarity to the single nearest neighbour (the closest replacement); $f_8 = 1 - f_7$ flips the sign so that isolated documents, those that cannot be easily replaced, score high.

Coverage (f_9–f_{11}). Coverage features quantify how many queries depend on a document as their primary retrieval target, drawing on greedy set-cover [16] and query-log-driven index pruning [24]. f_9 is the standard count of queries for which d is the nearest document (the Coverage baseline's sole signal). The two further features are novel in this paper: f_{10} adds the *confidence* dimension by averaging the margin between $\text{sim}(q, d)$ and the second-nearest similarity over the queries d tops; f_{11} adds the *fallback* dimension by counting queries for which d is the 2nd or 3rd nearest. Section 5.3 shows that the optimizer regularly assigns large weights to f_{10} and f_{11}, sometimes inverting the sign of f_9 entirely.

Diversity (f_{12}) and Relevance Anchoring (f_{13}). $f_{12} = 1 - f_6$ surfaces documents in sparse embedding regions, complementing f_8 so that the linear scorer can express either-end preferences without requiring negative weights. f_{13} is the only supervised feature: given

the union $\mathcal{D}_{rel} = \{d : \exists q \in \mathcal{Q}, (q, d) \in \text{qrels}^+\}$ of training-relevant documents, f_{13} scores each corpus document by its cosine similarity to $\bar{\phi}_{rel}$, the centroid of $\mathcal{D}_{rel}$. Conceptually, f_{13} is a one-dimensional projection of the labelled supervision; it is critical on small datasets where unsupervised signals saturate (Section 5.3).

These features span three signal types: unsupervised query-dependent ($f_1 - f_5$, $f_9 - f_{11}$), unsupervised corpus-geometry ($f_6 - f_8$, f_{12}), and supervised relevance-anchoring (f_{13}). Crucially, the query-dependent features are workload-aware by construction: f_9, f_4, f_5, and f_{10} directly measure how many training queries depend on each document and with what confidence, so training the linear scorer on them is an explicit optimisation for the query distribution that $\mathcal{Q}$ represents.

3.3 Optimization via Evolution Strategies

Having defined the linear score $s(d) = \mathbf{w}^\top \mathbf{f}(d)$ and the top-B selection it induces (Eq. 2), the remaining question is how to learn $\mathbf{w}$. Standard gradient-based optimisation is not directly applicable: a small change in $\mathbf{w}$ may swap two documents near the selection boundary, causing a discontinuous jump in nDCG@10. The objective in Eq. 1 is therefore non-differentiable with respect to $\mathbf{w}$, and this is by design, not a limitation to work around. The alternative would be to replace nDCG@10 with a differentiable surrogate and top-B selection with soft gating, as standard in learning-to-rank; but such relaxations decouple the training objective from the deployment objective and do not transfer to BM25 or cross-modal retrieval where no continuous scoring function exists (Sections 4.4, 4.5).

At each iteration t, ES estimates the gradient of the expected reward by evaluating perturbed weight vectors:

1. **Perturb**: Sample K noise vectors $\boldsymbol{\epsilon}_k \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})$, for $k = 1, \ldots, K$.
2. **Select**: For each perturbation, compute the document scores $(\mathbf{w}_t + \boldsymbol{\epsilon}_k)^\top \mathbf{f}(d_i)$ for all d_i and select the top-B documents.
3. **Evaluate**: Compute $R_k = \text{nDCG@10}(\mathcal{Q}_{\text{train}}, \mathcal{S}_k)$ by running retrieval over the selected subset.
4. **Normalize**: Compute advantages $A_k = (R_k - \bar{R})/(\text{std}(R) + \epsilon)$ to reduce variance.
5. **Update**:

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \frac{\alpha}{K\sigma} \sum_{k=1}^{K} A_k \, \boldsymbol{\epsilon}_k \tag{4}$$

The scalar R_k is the end-to-end retrieval reward under the perturbed selection, not a differentiable surrogate. The update in Eq. 4 is equivalent to REINFORCE on a Gaussian-perturbed parameter policy [31]; it differs from GRPO [33], which trains a stochastic token-level policy rather than a fixed-dimensional parameter vector. Section 5.2 verifies empirically that ES is necessary: evaluation-matched random search on the unit sphere reaches weaker and substantially less stable solutions, trailing ES by 5–7 pp at aggressive budgets with 2–10× higher seed-to-seed variance, isolating the value of the guided gradient estimate from the value of the feature framework itself.

4 Main Experiments

The goal of this section is to evaluate whether QDS provides a robust and scalable solution for budget-constrained index pruning. Specifically, we test whether query-aware document selection improves retrieval quality retention under fixed document budgets, whether the learned selection strategy is robust across retrievers and modalities, and whether the method remains effective at web scale. The experiments are organized to progressively stress the method: we first compare QDS against query-aware, query-agnostic, and supervised pruning baselines on standard IR benchmarks; then test robustness across dense encoders, BM25 retrieval, and non-text modalities; and finally evaluate scalability on a 120M-passage corpus.

4.1 Experimental Setup

Our setup tests three questions: (i) whether query-aware pruning improves retention under fixed indexing budgets, (ii) whether the learned signals transfer across retrieval paradigms and representation families, and (iii) whether the observed behaviours persist beyond textual retrieval.

Datasets. We evaluate on 10 corpora spanning five orders of magnitude in size, six text domains, and two non-text modalities (Table 2). Five BEIR [35] benchmarks (ArguAna, FiQA, SciDocs, SciFact, NFCorpus) range from 3.6K to 57.6K documents and cover argumentative, financial, scientific, and biomedical IR; these are standard for retrieval evaluation but operate well below production scale, a consideration we revisit in Section 5.4. HotpotQA [43] (5.2M), MS-MARCO v1 [4] (8.8M), and MSMARCO v2 [5] (120M, deduplicated from 138M) test larger corpora. MSCOCO [26] (123K image/caption pairs) and Clotho [14] (1,045 audio clips) test generalisation beyond text.

Data splits. We use native splits where available (FiQA, NFCorpus, HotpotQA); hold out 20% of training queries for SciFact (no dev set); create 70/15/15 splits for ArguAna, SciDocs (test only) and Clotho (audio-level); split MSMARCO-v1 training queries 90/10 for train/dev, using the official dev as test; and adopt the Karpathy[2] splits for MSCOCO. TREC DL-2022 queries [13] serve as test set for MSMARCO-v2. Features requiring relevance judgements are computed exclusively from the training partition; dev queries are used only for ES early stopping.

Indexing and embedding models. As dense retrievers we use MiniLM-L6-v2 [38] (33M, 384-dim), BGE [40] (109M, 768-dim), Qwen3-0.6B and Qwen3-4B [42], with MiniLM as default unless stated otherwise; MSCOCO and Clotho use CLIP ViT-B/32 [29] and CLAP-HTSAT-fused [39]. No retriever is fine-tuned.

Baselines. We compare against eight baselines grouped by the signal they use. *Query-aware heuristics (2)*: *Query Relevance* selects the highest f_1 (`max_query_sim`); *Coverage* selects the highest f_9 (`coverage_count`), a greedy set-cover heuristic. *Query-agnostic heuristics (5)*: *Random* samples uniformly; *Magnitude* [19] uses the L2 norm of the un-normalised embedding; *Dissimilarity* selects documents farthest from the corpus centroid; *kNN density* selects the densest embedding regions (k=10); *ITN* [45] uses the ratio of unique to total terms. *Supervised (1)*: *QualT5-zs* [11], the released `qt5-base` checkpoint (~220M parameters) trained on MSMARCO v1 and applied zero-shot everywhere; this is the strongest available model-based baseline (Section 2). The grouping disentangles three effects:

[2] https://huggingface.co/datasets/yerevann/coco-karpathy

Table 2: Dataset statistics. nDCG@10 with MiniLM-L6-v2 (text), CLIP ViT-B/32 (MSCOCO), and CLAP-HTSTAT (Clotho).

Dataset	#Docs	Train	Dev	Test	nDCG	Domain
ArguAna	8,674	984	212	212	36.6	Argument
FiQA	57,638	5,500	500	648	36.8	Finance
SciDocs	25,657	700	150	150	20.3	Scientific
SciFact	5,183	648	161	300	64.5	Biomedical
NFCorpus	3,633	2,590	324	323	31.6	Biomedical
MSMARCO v1	8.8M	452,646	50,293	6,980	36.5	Web
HotpotQA	5.2M	85,000	7,405	7,405	46.5	Multi-hop
MSMARCO v2	120M	452,646	50,293	76 (DL-22)	37.8	Web
MSCOCO	123,287	113,287	5,000	5,000	15.66	Image
Clotho	1,045	731	157	157	43.8	Audio

the contribution of query information, of learning to combine features rather than using one, and of a trained language model.

Training. We fix $\sigma = 0.3$, $\alpha = 0.1$, $K = 16$ a priori across all experiments, without per-dataset tuning; these are standard ES defaults [31] and halving or doubling σ changes final retention by <1 pp on BEIR. We train one weight vector per budget B for at most 2,000 iterations, checking dev nDCG@10 every 10 and retaining the best checkpoint; the dev plateau is typically reached within 200–500 iterations on BEIR. On large corpora (MSMARCO, HotpotQA), where evaluating all training queries per iteration is prohibitive, we use a rotating random subsample of 5,000 queries. Feature computation is a one-time offline cost: seconds on BEIR, ~30 minutes on MSMARCO v1 with a single GPU, and sharded across machines at 120M scale; ES optimisation then runs in 1–3 hours depending on corpus size.

Evaluation protocol. For each dataset we apply all methods at budgets $B \in \{10\%, 30\%, 50\%, 70\%, 90\%\}$ of the corpus size, retrieve the top-10 documents for each test query from the pruned index using the dense retriever (or BM25 where specified), and compute nDCG@10. We report *retention*, the performance of the pruned index relative to the full one: $\text{Ret}\% = \text{nDCG@10}_{\text{pruned}}/\text{nDCG@10}_{\text{full}} \times 100$, with full-corpus values in Table 2. Stochastic methods (Random, QDS) are run over 5 seeds and averaged; tables report a one-tailed Welch's t against the strongest non-QDS competitor at $p < 0.05$ (df= 4). We prefer cross-seed Welch to a per-query paired bootstrap because each seed is an independent realisation of the full pipeline (ES initialisation, query subsample, selection, retrieval), which captures the dominant variance source.

4.2 Main Results

We first evaluate whether query-aware document selection improves retrieval retention under fixed indexing budgets across standard IR benchmarks and web-scale retrieval settings. Table 3 reports per-budget retention on seven datasets with MiniLM-L6-v2 as dense retriever, spanning 3.6K to 8.8M documents. QDS achieves the best retention in 24 of 35 dataset-budget cells (19/25 on BEIR, all 5 on HotpotQA). The remaining cells concentrate on MSMARCO v1, where QualT5-zs dominates every method. This is the single dataset on which QualT5-zs was originally trained [11]: its advantage is therefore expected and does not transfer, as we show on the other 9 corpora here and on MSMARCO v2 in Section 4.6, where QDS leads at 120M-passage scale with the same architecture. We return

Table 3: Retention (%) with MiniLM-L6-v2. Bold denotes best results. $^{(*)}$: Welch's t, $p < .05$ vs. best non-QDS.

DS	Method	10%	30%	50%	70%	90%
ArguAna	Random	22.5 ± 3.1	50.9 ± 2.7	73.5 ± 3.6	85.9 ± 2.4	94.0 ± 1.9
	QueryRel.	0.8	65.7	87.1	99.2	99.4
	Coverage	0.0	103.9	104.9	102.8	101.1
	kNN-density	0.8	10.4	22.6	44.1	77.7
	Dissim.	13.5	36.5	52.6	73.9	87.5
	Magnitude	45.5	99.3	95.4	92.2	98.5
	ITN	43.5	81.2	99.7	110.9	101.0
	QualT5-zs	11.1	30.5	53.6	74.3	98.2
	QDS (Ours)	**60.5 ± 0.7***	**103.9 ± 1.8**	**119.0 ± 1.4***	**117.6 ± 1.9***	**114.2 ± 0.0***
FiQA	Random	19.0 ± 2.0	42.4 ± 1.8	62.5 ± 0.3	78.4 ± 0.6	94.0 ± 1.4
	QueryRel.	24.0	62.3	84.5	97.4	99.9
	Coverage	23.2	48.0	66.2	80.6	94.2
	kNN-density	7.0	30.6	57.9	80.1	94.9
	Dissim.	7.9	42.5	71.5	89.2	97.3
	Magnitude	20.3	45.6	66.2	79.5	94.0
	ITN	7.7	19.9	33.4	50.0	80.7
	QualT5-zs	25.8	54.4	74.6	87.3	98.7
	QDS (Ours)	**28.7 ± 0.2***	**67.6 ± 0.4***	**90.7 ± 0.2***	**98.5 ± 0.2***	**100.4 ± 0.2***
SciDocs	Random	28.0 ± 2.5	56.4 ± 5.7	72.7 ± 5.4	87.0 ± 2.3	95.7 ± 2.1
	QueryRel.	25.2	45.0	67.3	86.3	95.3
	Coverage	26.0	45.6	60.8	83.3	98.1
	kNN-density	4.0	24.7	53.4	74.2	89.8
	Dissim.	14.7	46.8	67.4	83.1	93.1
	Magnitude	23.8	46.3	71.5	79.3	93.6
	ITN	30.6	51.7	75.4	85.8	93.8
	QualT5-zs	**31.6**	48.5	62.0	77.4	96.1
	QDS (Ours)	31.2 ± 0.3	**58.3 ± 2.0***	**85.2 ± 0.8***	**96.7 ± 0.6***	**98.8 ± 0.6***
SciFact	Random	14.2 ± 2.0	36.7 ± 3.8	58.9 ± 5.8	75.6 ± 4.6	91.5 ± 3.0
	QueryRel.	34.5	54.6	68.9	86.6	95.8
	Coverage	45.3	61.8	**79.6**	**90.9**	**98.4**
	kNN-density	9.0	28.4	47.4	75.4	93.8
	Dissim.	8.0	33.6	53.0	73.3	90.7
	Magnitude	12.4	41.9	62.2	74.3	92.9
	ITN	13.1	32.0	52.8	72.8	88.7
	QualT5-zs	6.0	30.4	52.3	69.8	91.4
	QDS (Ours)	**47.0 ± 0.2***	**63.0 ± 0.6***	71.2 ± 0.1	89.5 ± 0.8	96.2 ± 1.4
NFCorpus	Random	32.8 ± 2.3	58.2 ± 1.6	72.7 ± 2.1	84.3 ± 0.9	95.2 ± 1.5
	QueryRel.	30.7	56.7	71.1	81.9	95.4
	Coverage	31.9	59.3	70.9	79.4	95.5
	kNN-density	22.3	55.1	71.0	**86.7**	**97.7**
	Dissim.	25.9	48.0	64.8	82.8	95.4
	Magnitude	21.6	43.6	71.8	83.9	95.4
	ITN	33.5	60.9	73.6	86.0	95.0
	QualT5-zs	28.7	55.5	72.7	83.5	93.5
	QDS (Ours)	**37.6 ± 0.6***	**62.2 ± 0.3***	**76.8 ± 0.8***	86.7 ± 0.3	95.4 ± 0.2
HotpotQA	Random	14.2 ± 0.7	36.9 ± 0.9	56.9 ± 0.4	74.8 ± 0.3	91.8 ± 0.2
	QueryRel.	52.3	69.3	80.8	90.4	97.8
	Coverage	59.5	74.6	81.5	89.1	96.5
	kNN-density	20.8	49.3	72.1	89.4	98.6
	Dissim.	24.9	54.0	74.0	87.8	96.7
	Magnitude	35.2	53.2	63.5	81.2	93.8
	ITN	8.1	25.8	46.3	70.2	93.6
	QualT5-zs	22.1	49.6	70.2	87.2	99.6
	QDS (Ours)	**65.2 ± 0.2***	**81.4 ± 0.6***	**91.9 ± 0.4***	**96.8 ± 0.3***	**99.8 ± 0.2***
MS v1	Random	11.2 ± 0.0	32.7 ± 0.7	52.7 ± 0.7	72.2 ± 0.6	90.5 ± 0.2
	QueryRel.	24.0	34.2	56.5	72.3	95.3
	Coverage	14.5	15.5	16.6	17.3	95.8
	kNN-density	9.1	32.0	56.5	77.9	95.6
	Dissim.	14.5	32.4	53.9	72.6	91.9
	Magnitude	5.5	7.6	9.1	74.3	72.2
	ITN	10.7	30.3	50.8	71.9	94.5
	QualT5-zs	**40.0**	**74.0**	**89.7**	**97.0**	**99.9**
	QDS (Ours)	27.8 ± 0.1	49.5 ± 0.3	68.0 ± 0.9	81.0 ± 0.1	96.9 ± 0.2

to this case below and provide a direct in-domain T5 comparison in Section 5.1.

We identified multiple key insights:

First, ArguAna [37] requires retrieving *counter-arguments*, thus semantically similar documents are often distractors rather than answers. At 10% budget, Coverage (0.0%) and Query-Relevance (0.8%) collapse and they retain the nearest neighbors that are systematically wrong. Magnitude and ITN (45.5, 43.5%) select textually rich documents regardless of their relevance for queries, and this richness allows them to achieve >90% retention in a budget of

Table 4: Average rank across 5 BEIR datasets × 5 budgets (25 cells per model; rank 1 = best retention).

Method	MiniLM	BGE	Qwen-0.6B	Qwen-4B	Avg
QDS (Ours)	**1.4**	**1.0**	**1.1**	**1.1**	**1.2**
QueryRel.	4.0	4.2	3.3	3.8	3.8
Coverage	4.6	4.2	4.1	5.1	4.5
ITN	5.2	4.8	4.7	4.8	4.9
QualT5-zs	4.2	5.6	5.1	4.7	4.8
Random	5.7	5.4	5.6	5.5	5.6
Magnitude	5.6	6.2	5.7	6.2	5.9
Dissim.	5.9	6.6	6.1	6.2	6.2
kNN-density	6.9	6.7	8.0	7.7	7.4

50%. QualT5-zs, despite being a 220M-parameter neural quality estimator, shows worse performance compared to baselines on this dataset. We hypothesise that this is due to its training on MS-MARCO, where the quality signal does not capture argumentative relevance. In contrast, QDS consistently achieves higher retention, exceeding 100% relative. The model learns a negative weight on the `coverage_count` feature, penalising relevant but too general documents that are otherwise incorrectly retrieved (see Section 5.3).

Second, the strongest competitor shifts across datasets: Query Relevance on FiQA, Coverage on HotpotQA, ITN on SciDocs and NFCorpus, while QDS is consistently first or second. QualT5-zs is competitive on FiQA (25.8 at 10%, second to QDS) but inconsistent: on HotpotQA it reaches only 49.6 at 30% vs. QDS's 81.4.

Third, SciFact pairs each claim with a single gold passage in a corpus of only 5183 documents. At this scale, hard negatives are sparse and Coverage reliably identifies gold passages, making it a near-optimal heuristic at ≥50% budget. QDS still wins at the aggressive 10/30% budgets.

MSMARCO v1 (8.8M passages) shares SciFact's one-positive-per-query structure but at three orders of magnitude more documents and 452K training queries, saturating the corpus with hard negatives—passages that are nearest neighbours to hundreds of queries yet relevant to none. Coverage-based selection cannot distinguish these from true positives and greedily fills the index with them, plateauing at 14–17% retention for budgets 10–70%, *worse than uniform random sampling*. QDS avoids this trap and beats every heuristic baseline (27.8 / 49.5 / 68.0 / 81.0 / 96.9), but does not match QualT5-zs (40.0 / 74.0 / 89.7 / 97.0 / 99.9). The gap is distributional, not structural: QualT5-zs was trained on MSMARCO v1 itself [11], giving it direct access to the relevance distribution that QDS's structural features must infer from query–document geometry alone. The advantage is dataset-specific: on the other 9 corpora QualT5-zs trails every heuristic-aware method (mean rank 4.8 across encoders, Table 4), and on MSMARCO v2 (120M passages) QDS overtakes it at 3 of 5 budgets (Section 4.6). We explore the gap further in Section 5.1, where re-training T5-base in-domain on BEIR qrels does not reverse the verdict elsewhere.

4.3 Robustness Across Embedding Models

We next test whether the learned retrieval-structural signals transfer across embedding architectures and representation scales. The results above use MiniLM-L6-v2 as the default retriever; here, we re-run all methods on the BEIR benchmarks with three additional embedding models: BGE-base (109M), Qwen3-Embedding-0.6B, and Qwen3-Embedding-4B. Each encoder defines its own retrieval space and full-corpus nDCG@10, against which retention is computed independently.

Table 4 summarises the outcome. For each dataset-budget pair (5 datasets × 5 budgets), we rank all 9 methods by retention from 1 (best) to 9 (worst) and report the average rank. QDS achieves a mean rank of 1.0-1.4 with every encoder, being always the best method on average. The next-best method, QueryRel., averages 3.8 and varies across models. QualT5-zs is competitive on MiniLM (4.2) but degrades on BGE (5.6) and Qwen-0.6B (5.1), confirming that its MSMARCO-trained signal does not transfer across encoders.

The average rank difference between QDS and all other systems indicates (i) QDS remains consistently robust across datasets, pruning budgets, and embedding models; (ii) all other methods are domain-dependent, ranking well on some configurations and poorly on others, with no single baseline maintaining a stable position across the board. QDS requires no domain-specific tuning and adapts to each configuration through its learned feature weights.

Beyond aggregate ranks, one qualitative pattern recurs across encoders: ArguAna's Less-is-More regime holds with all four models, with every encoder exceeding 100% retention at budgets ≥30%. The distractor-removal strategy is therefore a property of the task topology, not of the embedding space, a point we return to in Section 5.3, where the learned negative weight on `coverage_count` explains the mechanism.

4.4 BM25 Robustness

A natural concern with dense-retrieval-based pruning is that the learned selections may not transfer beyond dense embedding spaces. We evaluate this in two progressively stronger settings, both under Pyserini BM25 [25, 30] (k_1=0.9, b=0.4). First, we directly apply the dense-trained QDS weights to BM25 retrieval (*dense→BM25 transfer*). Second, we recompute all features from BM25 query-document interactions and re-optimise the weights entirely under sparse retrieval (*BM25-native*). Table 5 reports retention on all five BEIR datasets. For reference, the dense-setting baselines are included in the same table, while BM25-native win counts discussed below are computed against BM25-recomputed baselines. Full per-dataset comparisons are included in the code release.

Transfer test: dense selections under BM25. We take QDS's selections trained from MiniLM-L6-v2 dense features and re-evaluate them with BM25 instead of dense retrieval. Nothing changes except the evaluation retriever. QDS wins or ties the best baseline in 18 of 25 cells, and ArguAna's Less-is-More effect strengthens, peaking at 126.9% at 30% vs. 119.0% dense. Where QDS loses, the winners are BM25-aligned heuristics, such as ITN on ArguAna at 70%, where a term-frequency retriever naturally rewards lexically diverse passages, and Coverage on SciFact (the same one-to-one relevance pattern as in the dense setting). This confirms that the structural signal QDS learns is not an artefact of the dense embedding space.

Native test: BM25 features for training, BM25 evaluation. The transfer test still computes features from dense embeddings. A stronger question is whether QDS works with *no dense embeddings at all*. We re-instantiate the entire framework under BM25: all 13 features are recomputed from lexical query–document scores (`rank_bm25`) and TF-IDF cosine for document–document similarity,

Table 5: BM25 evaluation on BEIR. Two QDS rows per dataset: dense→BM25 and BM25-native; baselines from the dense setting. Bold: best per cell; * as in Table 3.

DS	Method	10%	30%	50%	70%	90%
ArguAna	Random	24.1 ± 2.1	52.1 ± 4.5	74.3 ± 7.0	85.5 ± 5.2	93.8 ± 2.2
	QueryRel.	1.0	69.6	87.7	98.9	98.9
	Coverage	0.0	106.3	107.7	104.6	101.9
	kNN-density	49.3	84.6	98.6	98.7	100.2
	Dissim.	14.2	47.3	69.7	81.0	95.4
	Magnitude	28.8	50.9	69.1	84.4	90.2
	ITN	47.7	94.1	111.4	127.8	106.0
	QualT5-zs	13.6	29.1	54.8	76.3	100.0
	QDS (dense → BM25)	63.8 ± 5.1*	**126.9 ± 1.8***	125.6 ± 1.6*	123.3 ± 1.5	**119.9 ± 0.0***
	QDS (BM25-native)	**65.9 ± 1.9**	118.3 ± 1.5	**131.9 ± 0.3**	**133.5 ± 0.4**	118.9 ± 0.3
FiQA	Random	20.5 ± 2.6	45.8 ± 3.9	65.3 ± 3.7	79.2 ± 1.4	93.6 ± 1.8
	QueryRel.	21.9	55.9	81.9	96.1	100.0
	Coverage	21.5	48.3	65.4	79.2	93.6
	kNN-density	16.3	36.8	61.1	81.0	95.7
	Dissim.	9.2	44.5	72.2	88.6	96.4
	Magnitude	20.0	48.0	63.6	76.5	95.8
	ITN	6.8	17.7	29.2	47.1	81.5
	QualT5-zs	28.5	53.4	74.2	88.1	98.4
	QDS (dense → BM25)	26.0 ± 0.9	60.3 ± 0.1*	85.5 ± 0.2*	97.8 ± 0.3*	99.5 ± 0.1
	QDS (BM25-native)	**44.7 ± 0.4**	**85.7 ± 0.6**	**97.1 ± 0.4**	**99.1 ± 0.1**	**100.1 ± 0.0**
SciDocs	Random	25.6 ± 5.6	58.8 ± 5.7	71.8 ± 5.3	88.3 ± 3.4	97.4 ± 1.2
	QueryRel.	18.6	41.3	68.1	90.6	95.4
	Coverage	19.3	39.6	58.0	83.4	100.0
	kNN-density	**34.0**	59.1	75.3	89.0	**100.4**
	Dissim.	15.7	45.2	65.7	78.3	90.2
	Magnitude	22.7	57.0	66.3	85.8	95.0
	ITN	27.3	44.9	72.5	87.7	95.2
	QualT5-zs	29.1	60.2	75.7	87.9	93.9
	QDS (dense → BM25)	27.4 ± 0.3	53.8 ± 1.0	**83.8 ± 2.8***	**92.7 ± 1.4***	97.9 ± 0.4
	QDS (BM25-native)	28.2 ± 1.8	**62.5 ± 1.6**	80.4 ± 0.2	90.1 ± 1.2	99.3 ± 0.5
SciFact	Random	13.4 ± 1.9	35.4 ± 2.7	57.5 ± 4.9	74.5 ± 4.2	91.1 ± 2.5
	QueryRel.	32.4	53.2	69.0	85.2	95.0
	Coverage	42.9	61.8	79.7	90.8	99.1
	kNN-density	10.0	29.8	57.1	77.2	92.3
	Dissim.	8.0	33.2	53.2	74.2	92.0
	Magnitude	16.6	38.8	59.1	76.5	92.1
	ITN	10.9	29.7	51.7	72.1	89.6
	QualT5-zs	5.4	29.2	51.5	68.6	91.1
	QDS (dense → BM25)	45.3 ± 0.1*	58.7 ± 0.2	71.6 ± 0.1	86.8 ± 0.6	95.3 ± 0.8
	QDS (BM25-native)	**54.4 ± 0.4**	**68.9 ± 0.2**	**86.4 ± 0.2**	**93.9 ± 0.2**	**99.4 ± 0.5**
NFCorpus	Random	28.6 ± 4.0	55.1 ± 4.1	70.3 ± 3.6	83.1 ± 1.6	94.2 ± 1.4
	QueryRel.	25.7	52.7	67.9	82.1	93.9
	Coverage	27.7	57.2	69.9	78.5	96.1
	kNN-density	22.7	44.9	64.0	76.9	93.5
	Dissim.	17.7	41.4	63.5	81.4	95.9
	Magnitude	28.8	55.2	72.3	88.2	95.4
	ITN	28.1	57.9	70.8	86.4	97.3
	QualT5-zs	22.8	54.7	71.2	86.0	96.1
	QDS (dense → BM25)	32.8 ± 0.6*	59.9 ± 0.2*	76.0 ± 0.6*	86.8 ± 0.2	93.0 ± 0.4
	QDS (BM25-native)	**38.8 ± 0.7**	**65.9 ± 0.4**	**82.4 ± 0.1**	**90.6 ± 0.1**	**98.9 ± 0.5**

and ES trains on a BM25 nDCG reward.[3] Baselines are also recomputed from BM25 scores, so the comparison is end-to-end lexical. QDS wins 23 of 25 cells (vs. 18 in the transfer setup), and ArguAna's Less-is-More peaks at 133.5%. On SciFact, where the dense-transfer variant lost to Coverage at ≥30%, the native variant leads at every budget. The 13 structural slots accommodate a fully lexical instantiation: the signals QDS captures are retrieval-structural, not embedding-specific.

4.5 Multi-Modal Transfer: Image and Audio Retrieval

The 13 QDS features depend only on query–document interactions and corpus geometry in the retrieval space; nothing in the formulation is specific to textual retrieval. To test whether the learned pruning signals transfer across modalities, we apply QDS to text-to-image retrieval on MSCOCO [26] using CLIP ViT-B/32 [29], and to text-to-audio retrieval on Clotho [14] using CLAP-HTSAT-fused [39]. Media items serve as "documents" and captions as "queries"; the only change is the encoder backbone.

QualT5-zs and ITN are omitted because they rely on modality-specific textual signals that do not transfer naturally to image or audio retrieval. Table 6 reports retention across five budgets.

On MSCOCO, QDS wins at every budget, leading Coverage by 15 pp at 50% (87.0 vs. 72.1) and entering the Less-is-More regime at ≥70% alongside Coverage as both benefit from MSCOCO's redundancy (five captions per image). On Clotho, QDS wins 3 of 5 budgets (30/50/70%) by 4–6 pp over the strongest heuristic; at 10%, Dissimilarity edges QDS on a 1,045-clip corpus where variance is high reflecting the small corpus size (1,045 clips). The relevance patterns mirror the text setting: Coverage is competitive on MSCOCO's one-to-one structure (as on SciFact), and `coverage_margin` saturates near +1 on both cross-modal datasets (Section 5.3). To our knowledge, QDS is the first learned, query-aware index-pruning method that operates uniformly across text, image, and audio retrieval.

Table 6: Cross-modal retention (%) on MSCOCO (CLIP ViT-B/32) and Clotho (CLAP-HTSAT-fused); only the encoder changes. Bold: best per cell; * as in Table 3.

Dataset	Method	10%	30%	50%	70%	90%
MSCOCO	Random	23.6 ± 1.4	48.3 ± 1.6	65.9 ± 1.2	82.9 ± 1.8	95.1 ± 0.4
	QueryRel.	20.0	46.9	65.9	81.6	95.1
	Coverage	28.2	61.5	72.1	101.9	100.6
	kNN-density	5.7	21.9	41.4	63.6	87.9
	Dissim.	17.1	41.5	63.8	83.0	95.3
	Magnitude	24.4	48.9	65.9	79.5	93.5
	QDS (Ours)	**28.1 ± 1.2**	**62.0 ± 0.4***	**87.0 ± 0.7***	**102.0 ± 0.5**	**106.4 ± 0.2***
Clotho	Random	12.0 ± 8.9	31.6 ± 22.4	44.7 ± 31.6	54.7 ± 38.8	75.8 ± 28.3
	QueryRel.	10.1	29.2	48.4	80.6	91.3
	Coverage	8.2	35.1	57.1	75.4	92.0
	kNN-density	16.7	34.9	51.0	73.4	91.5
	Dissim.	**21.9**	41.8	65.4	79.8	**97.5**
	Magnitude	15.5	42.5	64.2	82.7	91.9
	QDS (Ours)	15.8 ± 0.0	**46.8 ± 0.9***	**68.3 ± 1.3***	**84.2 ± 2.9**	96.7 ± 1.3

Table 7: MSMARCO v2 (120M passages), TREC DL-2022 (76 queries), MiniLM-L6-v2. Full-corpus nDCG@10 = 0.378.

Method	10%	30%	50%	70%	90%
Random	40.1	70.1	82.5	93.8	98.1
Coverage	53.3	67.3	78.3	90.2	96.8
Query Rel.	47.9	84.1	94.2	97.7	99.7
kNN-density	10.5	27.4	50.4	69.4	89.2
Dissim.	42.6	75.6	84.2	94.6	100.6
Magnitude	41.7	64.0	79.3	90.3	96.8
ITN	35.3	54.5	68.5	82.9	94.3
QualT5-zs	**63.2**	80.5	92.2	99.2	**100.8**
QDS (Ours)	61.2	**85.7**	**96.2**	**101.3**	100.0

4.6 Scalability to 120M Documents

To verify that QDS scales beyond single-GPU corpora, we evaluate on MSMARCO v2 [5] (120M passages, deduplicated from 138M) with TREC DL-2022 [13] queries (Table 7). Feature computation is sharded across 16 partitions on 4 machines with no cross-shard communication: each shard computes its 13 features independently, and ES optimisation runs on the concatenated feature matrix as on any other dataset. Due to the computational cost of evaluating

[3] We use `rank_bm25` for the in-memory score matrix ES requires at training throughput, with the same Porter stemmer and stopword list as Pyserini. Final test evaluation is always Pyserini.

120M passages per perturbation, we report a single ES run and omit significance markers from Table 7; at inference, the learned weights score documents in microseconds per shard.

Table 7 reports retention across five budgets. QDS leads at 30–70% budget, with the largest margin at aggressive pruning: at 10% budget (a 10× index reduction), QDS retains 61.2% vs. 53.3% for Coverage and 40.1% for Random. At 30%, QDS leads Query Relevance by +1.6 pp (85.7 vs. 84.1) and QualT5-zs by +5.2 pp (85.7 vs. 80.5). At 70% QDS exceeds the full index (101.3%), confirming the Less-is-More effect at 120M scale. QualT5-zs leads at 10% (63.2 vs. 61.2) and 90% (100.8 vs. 100.0), but unlike on MSMARCO v1 where it dominates every budget, on v2 QDS wins 3 of 5 budgets, suggesting that the v1 gap reflects distributional proximity to QualT5's training data rather than a fundamental limitation of structural features. The framework generalises across five orders of magnitude in corpus size (1K to 120M) without architectural or hyperparameter change.

5 Generalization Analysis

Section 4.2 showed that QualT5-zs leads on MSMARCO v1, the dataset on which it was trained, but trails QDS elsewhere. This raises a generalisation question: is QualT5-zs's MSMARCO v1 lead the property of a stronger neural quality estimator, or an artefact of in-distribution training data? We analyse this and three further generalisation properties of the framework: (i) whether re-training a 220M-parameter T5 classifier in-domain on a different corpus preserves its MSMARCO advantage (Section 5.1), (ii) which QDS design choices drive the observed gains (Section 5.2), (iii) what selection strategies the optimiser learns and how they relate to dataset structure (Section 5.3), and (iv) how QDS's retention varies across queries of decreasing similarity to the training workload (Section 5.4).

5.1 In-Domain T5-Base Comparison

Section 4.2 attributed QualT5-zs's MSMARCO v1 lead to its training-set exposure rather than to a fundamental advantage of neural quality estimators. We test this directly by re-training T5-base (∼220M parameters) in-domain on two BEIR collections, ArguAna and SciDocs, chosen as representatives of adversarial and standard retrieval. We follow Chang et al.'s prompt and head reduction exactly, training with binary cross-entropy on corpus-level qrels (every passage positive in any training qrel is labelled 1, the rest 0) for 5 seeds.[4]

Table 8 reports the comparison. QDS wins 9 of 10 cells. The single T5-favourable cell (ArguAna 30%) falls inside T5's ±6.3 pp seed envelope and is not distinguishable from a tie. Beyond mean retention, the *stability gap* is striking: T5-base shows ±6–10 pp variance on ArguAna at low budgets, while QDS stays within ±0.0–2.0 pp. A 13-parameter linear scorer thus yields both better mean performance and more reproducible selection than a 220M-parameter neural classifier given the same target-domain labels.

[4]This approximates Chang et al.'s protocol: they train on MSMARCO query-grouped triples, which BEIR does not natively provide.

Table 8: QDS vs. QualT5 trained in-domain on BEIR qrels (5 seeds). Bold: best per cell; * as in Table 3.

Dataset	Method	10%	30%	50%	70%	90%
ArguAna	QualT5-tr	46.7 ± 10.3	108.2 ± 6.3	107.5 ± 1.3	104.4 ± 0.5	101.3 ± 0.2
	QDS	**60.5 ± 0.7***	103.9 ± 1.8	**119.0 ± 1.4***	**117.6 ± 1.9***	**114.2 ± 0.0***
SciDocs	QualT5-tr	27.1 ± 2.2	52.4 ± 3.0	77.6 ± 2.6	92.3 ± 1.9	97.3 ± 1.3
	QDS	**31.2 ± 0.3***	**58.3 ± 2.0***	**85.2 ± 0.8***	**96.7 ± 0.6***	**98.8 ± 0.6***

5.2 Algorithm Design

Having shown that a 13-parameter linear scorer outperforms a 220M-parameter in-domain neural classifier, we turn to three natural objections about QDS's own configuration: that the linear scorer cannot capture feature interactions, that it ignores neural quality signals, and that Evolution Strategies may be unnecessary for optimising only 13 parameters. We test all three on the five BEIR datasets with MiniLM-L6-v2 (Table 9). *QDS+MLP* replaces the linear head with a 32-hidden-unit MLP over the same 13 features. *QDS+QualT5-feat* adds the QualT5-zs passage-quality score as a 14$^{\text{th}}$ feature to the linear scorer, testing whether a neural quality signal complements the structural ones. *QDS-RS* replaces ES with random search on the unit sphere S^{12} under equal compute budget: at each of $K \times T$ iterations trials we sample a unit vector $\mathbf{w} \sim \mathcal{N}(0, \mathbf{I})/\|\cdot\|$, score documents, evaluate nDCG@10 on the top-B selection, and retain the best-scoring $\mathbf{w}$ on the dev set. This isolates the value of the guided gradient estimate from the value of the feature framework itself.

Across all 25 cells the average gain over base QDS is +0.6 pp for QDS+MLP, −0.4 pp for QDS+QualT5-feat, and −4.6 pp for QDS-RS. The first two extensions fluctuate in both directions with no systematic pattern and remain within seed-to-seed standard deviations; the third reveals a clear gap. Random search trails ES by a median +5.2 pp at 10% budget, +7.2 pp at 30%, and +4.7 pp at 50%, collapsing to +1.5 pp at 90% as the problem becomes underconstrained and most weight vectors yield comparable selections. Beyond mean retention, ES exhibits 2–10× lower seed-to-seed variance (e.g. FiQA 30%: ±0.4 vs ±2.3; SciDocs 50%: ±0.8 vs ±6.4), mirroring the stability gap against in-domain T5 in Section 5.1.

This decomposes the QDS contribution along two axes. The 13-feature framework alone, optimised even with weak random search, already beats every single-feature heuristic by 5–15 pp at aggressive budgets (cf. Coverage, QueryRel., ITN in Table 3); ES then adds a further 5–7 pp by exploiting the local structure of the reward landscape that random sampling cannot capture. The base configuration thus sits at a sweet spot: simpler optimisation loses 5–7 pp at aggressive budgets, while more expressive scoring or richer features do not consistently help. We retain the linear scorer trained with ES: 13 parameters are sufficient, feature interactions do not help on average, neural signals do not complement the structural ones, the guided gradient estimate is necessary precisely in the aggressive-pruning regime where selection matters most, and each weight remains directly interpretable.

5.3 Learned Strategies, Ablation

Because QDS uses only 13 linear weights, the learned selection strategy is directly readable per dataset. Figure 1 shows the normalised weights at 30% budget for four representative tasks. The

Table 9: Design-choice ablations on BEIR (MiniLM-L6-v2, 5 seeds). Bold: best per cell.

DS	Variant	10%	30%	50%	70%	90%
ArguAna	QDS	**60.5 ± 0.7**	103.9 ± 1.8	119.0 ± 1.4	**117.6 ± 1.9**	**114.2 ± 0.0**
	QDS+MLP	59.4 ± 4.4	120.1 ± 1.2	118.8 ± 1.5	117.6 ± 0.8	114.1 ± 0.3
	QDS+QualT5-feat	53.8 ± 2.5	**122.2 ± 1.8**	**119.2 ± 1.4**	115.9 ± 1.3	114.0 ± 0.1
	QDS-RS	55.3 ± 3.8	96.7 ± 5.0	104.6 ± 3.8	113.0 ± 3.8	111.1 ± 1.0
FiQA	QDS	28.7 ± 0.2	67.6 ± 0.4	90.7 ± 0.2	98.5 ± 0.2	100.4 ± 0.2
	QDS+MLP	**29.4 ± 0.3**	65.3 ± 0.6	90.7 ± 0.1	**98.7 ± 0.2**	**100.5 ± 0.5**
	QDS+QualT5-feat	26.7 ± 2.3	**69.1 ± 1.1**	**92.6 ± 0.4**	98.6 ± 0.5	100.0 ± 0.3
	QDS-RS	25.3 ± 3.0	59.8 ± 2.3	86.0 ± 0.7	97.3 ± 0.2	99.1 ± 0.6
SciDocs	QDS	31.2 ± 0.3	**58.3 ± 2.0**	85.2 ± 0.8	**96.7 ± 0.6**	98.8 ± 0.6
	QDS+MLP	29.5 ± 1.3	52.1 ± 0.8	**85.9 ± 0.9**	96.6 ± 1.1	**99.0 ± 0.2**
	QDS+QualT5-feat	**31.6 ± 0.7**	55.3 ± 2.3	78.2 ± 1.6	85.7 ± 2.0	97.5 ± 0.2
	QDS-RS	29.1 ± 5.9	53.2 ± 4.7	78.0 ± 6.4	91.6 ± 2.5	95.9 ± 0.9
SciFact	QDS	**47.0 ± 0.2**	63.0 ± 0.6	71.2 ± 0.1	**89.5 ± 0.8**	96.2 ± 1.4
	QDS+MLP	53.4 ± 0.1	**65.7 ± 0.6**	**76.2 ± 2.2**	87.5 ± 1.6	**97.9 ± 0.3**
	QDS+QualT5-feat	46.4 ± 0.2	62.1 ± 0.3	70.2 ± 1.4	85.8 ± 1.0	95.3 ± 0.9
	QDS-RS	40.5 ± 1.7	53.4 ± 2.4	68.4 ± 3.4	81.8 ± 2.7	94.7 ± 0.6
NFCorpus	QDS	37.6 ± 0.6	**62.2 ± 0.3**	76.8 ± 0.8	86.7 ± 0.3	95.4 ± 0.2
	QDS+MLP	**39.9 ± 0.3**	62.2 ± 0.4	**77.5 ± 0.1**	**87.8 ± 0.6**	**96.7 ± 0.3**
	QDS+QualT5-feat	36.4 ± 0.9	61.5 ± 0.4	74.7 ± 0.6	86.3 ± 0.9	94.6 ± 0.8
	QDS-RS	31.8 ± 0.4	57.9 ± 2.0	72.3 ± 2.0	85.5 ± 0.6	95.3 ± 0.2

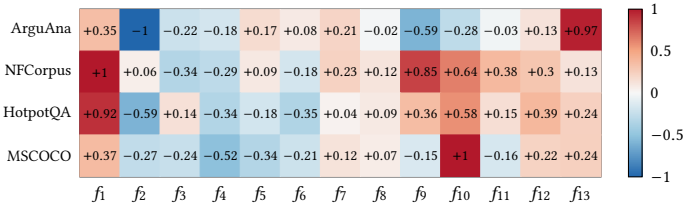

Figure 1: Normalised QDS weights at 30% budget for four representative tasks (blue: push out, red: pull in).

overarching finding is that the optimiser discovers strategies driven by *task topology*, not by media type or corpus scale: ArguAna (text) and MSCOCO (image) cluster together, as do NFCorpus (text) and Clotho (audio). Three weight patterns recur across all 8 datasets tested.

Anti-popular vs. pro-coverage. On ArguAna, where nearest neighbours are distractors, the optimiser assigns a negative weight to `coverage_count` (f_9) and a strongly negative weight to `mean_query_sim` (f_2), learning to push out documents that are generically similar to many queries. On NFCorpus and HotpotQA, both features receive positive weights: coverage is helpful when nearest neighbours are genuinely relevant.

Hard-negative detection via f_2. A negative weight on `mean_query_sim` appears on every dataset where hard negatives are prevalent (ArguAna, SciFact, HotpotQA, MSCOCO) and stays near zero elsewhere. Documents with high f_2 are nearest neighbours to many queries but relevant to none, the same class that drives Coverage's collapse on MSMARCO. The optimiser learns to detect and remove them without explicit hard-negative labels.

Coverage margin on one-to-one tasks. `coverage_margin` (f_{10}) saturates near +1 on MSCOCO, where each caption uniquely identifies one media item and the margin to the runner-up is the cleanest relevance discriminator. The same pattern appears more weakly on SciFact (one passage per claim) and HotpotQA (clustered bridges).

Ablation validates the weight patterns. Table 10 reports the leave-one-out Δ retention at 30% budget for BEIR datasets. The features that matter most in the ablation are precisely those with the largest weights in Figure 1: dropping `mean_query_sim` (f_2) costs −16.2 pp on ArguAna, where it carries the strongest negative weight; dropping `relevant_doc_sim` (f_{13}) costs −15.6 pp on ArguAna, where it saturates at +0.97; dropping `redundancy_max` (f_7) costs −14.1 pp on SciDocs. No single feature is critical across *all* datasets, the most impactful feature shifts with the task, which is why a learned combination outperforms any single heuristic. The supervised feature f_{13} dominates only on small-train datasets (ArguAna: 984 queries) and is negligible on larger ones (FiQA: +0.9 pp), consistent with the near-zero weight the optimiser assigns it there. Robustness tracks training-set size: FiQA and NFCorpus (≥2.6K queries) keep max $|\Delta| \leq 3.5$ pp; ArguAna and SciDocs (≤984) lose >14 pp when a key feature is removed.

Table 10: Leave-one-out ablation: Δ retention (%) at 30% budget (MiniLM-L6-v2). Negative = feature was helpful. Bold: $|\Delta| > 3$.

Dropped feature	ArguAna	FiQA	SciDocs	SciFact	NFCorpus
`max_query_sim`	**+4.0**	+0.4	**+7.0**	−0.6	+2.7
`mean_query_sim`	**−16.2**	−1.1	+0.8	−0.0	−0.0
`top_k_query_sim`	−0.2	−0.9	**−3.6**	+0.1	−0.6
`retrievability`	+0.6	+0.1	−0.2	+0.3	−0.2
`weighted_ret`	+0.1	−1.3	−1.9	+0.3	+0.4
`redundancy_mean`	−1.5	+0.0	−1.6	+0.8	−0.4
`redundancy_max`	+1.1	+0.1	**−14.1**	+0.4	−0.9
`isolation`	−0.4	+0.1	+0.3	+1.4	−0.0
`coverage_count`	+0.4	−0.2	−0.3	+0.1	−0.1
`coverage_margin`	−0.6	−1.9	−0.8	+0.8	+1.3
`backup_count`	−0.9	+0.1	**−5.3**	+1.8	**−3.5**
`uniqueness`	+0.4	+0.1	+0.5	+0.3	−0.0
`relevant_doc_sim`	**−15.6**	+0.9	+1.6	+0.8	+0.4

5.4 Generalisation Across Query Frequency: Head/Torso/Tail

QDS is trained on a fixed query set Q that proxies the expected workload. A natural question is whether the resulting selection generalises beyond the most common query patterns in Q, or whether retention concentrates on queries that look like training and collapses on rarer ones. We test this in two respects: *(i)* QDS should retain common-topic queries better than rare-topic queries (workload-awareness is an explicit design property), but *(ii)* it should not produce a degenerate selection that gives up on the rare tail.

For each test query we compute its mean cosine similarity to the k=10 nearest training queries and split the test set into tertiles: **head** (top 33%, queries on densely-covered training topics), **torso** (middle), and **tail** (bottom 33%, rarest topics relative to training). The same selections produced in Section 4.2 are re-evaluated per bucket against the corresponding per-bucket full-corpus nDCG@10. We caveat that on the smaller BEIR corpora the tail bucket is populated by queries whose ground-truth document is intrinsically distinctive (NFCorpus, SciDocs, SciFact, all <26K documents) and therefore well-retrieved even with minimal training support; the strongest evidence comes from FiQA (57K), HotpotQA (5.2M), and ArguAna where the adversarial structure removes the confound.

Table 11 reports head and tail retention per method across all five budgets on six text-retrieval datasets. Two findings emerge.

Workload-awareness pays off where pruning is aggressive. The gap between head and tail retention is largest at aggressive budgets and shrinks as more of the corpus is retained. Across the six datasets, QDS's head retention exceeds its tail by a median

of +29 pp at 10% budget and only +4 pp at 90%, indicating that workload-aware selection contributes precisely where it matters most. ArguAna is the exception: the head–tail gap stays in the +17 to +25 pp range at every budget (e.g. 30%: head 131.2, tail 107.2; 90%: head 126.3, tail 109.2), because distractor removal benefits both buckets but the head more strongly. On SciDocs at 50% the pattern inverts (head 89.6, tail 97.3) due to an absolute-difficulty confound: SciDocs tail queries are niche citation requests that the full corpus already retrieves with high nDCG@10, so the head's relative gain shrinks even when both buckets are well-preserved.

No tail collapse. A natural concern is whether QDS's head advantage comes at the expense of rare-query retention. Table 11 shows that this is not the case. On FiQA (57K documents), QDS wins both head and tail at 30, 50, and 70% budgets (e.g. 30%: head 87.9, tail 54.2 vs. next-best 53.9); on HotpotQA (5.2M documents), QDS leads both buckets at 10, 30, 50, and 90% budgets, with margins reaching +22 pp on tail at 30% (70.5 vs. Coverage 63.2). MSMARCO v1 (8.8M passages) tells the most revealing story: QDS retains 32.8 pp of tail at 30% budget and 52.6 pp at 50%, while Coverage stays below 7 pp on the tail at every budget under 90% and QueryRel. retains only 15.2 and 29.8 pp at the same budgets. QualT5-zs is the one method that leads both buckets on MSMARCO v1 (tail 73.1 pp at 30%), but it benefits from training directly on MSMARCO v1 itself (Section 5.1); among methods without that in-distribution access, QDS is the only one that retains a balanced head/tail profile rather than collapsing on the rare side. QDS therefore avoids the failure mode that defines Coverage and QueryRel. at web scale (Coverage MSMARCO v1 tail at 50%: 6.0 vs. QDS 52.6; QueryRel. tail at the same budget: 29.8): the workload-aware signal shifts the operating point along the Pareto frontier without producing a degenerate solution on either end. Where another method edges QDS on tail, typically Coverage on SciFact, or QualT5-zs on small-corpus NFCorpus and SciDocs (<26K documents, where the tail bucket is intrinsically easy), the same baselines concede 8–32 pp on the head.

Table 11: Retention (%) on head and tail buckets across budgets; tertiles by mean cosine similarity to the 10 nearest training queries. Bold: best per cell.

		10% budget		30% budget		50% budget		70% budget		90% budget	
Dataset	Method	Head	Tail	Head	Tail	Head	Tail	Head	Tail	Head	Tail
ArguAna	Random	11.2	25.4	26.2	55.5	59.0	80.7	73.2	91.0	83.7	96.0
	QueryRel.	0.0	0.0	85.4	33.3	99.9	68.8	99.8	98.0	99.8	98.5
	Coverage	0.0	0.0	96.4	**107.7**	101.2	108.1	100.9	104.7	100.3	101.7
	QualT5-zs	5.4	14.3	40.6	21.7	57.3	56.6	74.3	65.0	98.4	94.5
	QDS	**58.0**	**33.0**	**131.2**	107.2	**134.8**	**109.7**	**132.0**	**106.5**	**126.3**	**109.2**
NFCorpus	Random	44.0	20.5	71.1	49.7	84.1	64.0	94.5	77.6	94.6	**97.8**
	QueryRel.	48.5	19.0	75.4	39.7	**87.5**	51.7	**94.8**	69.1	99.3	86.7
	Coverage	42.3	22.9	67.7	50.8	78.3	66.4	87.0	80.3	98.6	94.6
	QualT5-zs	34.9	25.8	58.4	**61.9**	74.3	**73.4**	83.5	**86.0**	93.4	96.4
	QDS	**50.9**	**27.5**	**75.6**	49.5	84.9	64.6	91.6	75.9	**99.6**	79.2
SciFact	Random	10.9	16.4	39.3	**52.2**	68.8	74.1	80.0	80.9	93.9	92.6
	QueryRel.	60.2	13.1	87.3	24.0	89.9	51.4	98.2	69.9	100.6	88.3
	Coverage	71.0	**22.8**	82.5	44.2	**92.6**	**75.2**	99.9	**90.7**	100.3	**101.0**
	QualT5-zs	2.7	10.8	29.9	28.6	48.0	58.6	58.6	83.3	92.9	95.8
	QDS	**73.1**	22.5	**89.6**	43.6	91.1	58.1	**99.9**	74.5	**102.6**	94.2
SciDocs	Random	22.2	**36.0**	56.8	**68.5**	75.8	74.4	82.8	85.9	95.8	95.2
	QueryRel.	52.7	9.1	**84.2**	19.5	**95.2**	43.9	**101.2**	73.4	99.8	90.9
	Coverage	**55.1**	17.5	54.7	44.4	84.3	52.0	93.7	89.7	**102.7**	96.6
	QualT5-zs	39.7	16.3	57.8	64.4	76.5	76.7	92.5	80.0	92.8	86.7
	QDS	48.8	15.5	82.1	45.8	89.6	**97.3**	96.8	**94.3**	99.8	**97.1**
FiQA	Random	28.1	15.4	48.3	34.8	67.8	53.5	80.0	72.2	92.4	93.2
	QueryRel.	**51.2**	11.6	86.2	45.4	**98.1**	71.3	**100.0**	94.3	**100.0**	99.8
	Coverage	42.2	17.4	56.3	43.5	71.2	64.5	82.9	75.5	93.4	93.2
	QualT5-zs	31.6	**24.6**	54.0	53.9	72.8	75.0	83.7	89.9	96.0	99.1
	QDS	43.6	22.7	**87.9**	**54.2**	97.8	**82.0**	99.9	**97.8**	**100.0**	**100.2**
HotpotQA	Random	14.8	13.3	37.4	34.9	58.6	55.6	75.7	74.3	91.5	92.1
	QueryRel.	79.6	29.7	90.8	48.4	95.7	65.2	**98.4**	80.8	99.7	95.2
	Coverage	78.5	44.2	89.7	63.2	92.4	72.6	95.7	83.8	99.0	94.3
	QualT5-zs	15.8	14.2	40.0	37.4	60.6	55.8	76.2	74.3	92.0	91.5
	QDS	**86.3**	**44.8**	**94.0**	**70.5**	**96.8**	**85.0**	98.2	**95.3**	**99.8**	**98.8**
MS v1	Random	24.0	15.5	54.6	41.4	70.7	59.9	85.2	77.2	94.5	92.0
	QueryRel.	60.3	5.1	88.4	15.2	98.4	29.8	**96.0**	56.4	100.0	89.9
	Coverage	35.8	5.2	38.8	5.5	40.3	6.0	40.8	6.9	98.7	93.9
	QualT5-zs	49.6	**38.2**	**78.9**	**73.1**	**92.3**	**88.4**	98.0	**96.4**	100.0	**99.8**
	QDS	**54.5**	10.5	76.3	32.8	90.1	52.6	97.4	69.4	**100.1**	96.7

6 Conclusion

We introduced QDS, a query-aware pruning framework that scores documents using 13 retrieval-structural features combined through an ES-trained linear scorer that directly optimises nDCG@10 under fixed indexing budgets. Across dense, sparse, and multimodal retrieval settings, four findings emerge. First, optimal pruning is strongly dataset-dependent, and no single heuristic transfers reliably across retrieval regimes. Second, coverage-based selection degrades sharply at web scale by over-selecting high-frequency hard negatives. Third, lightweight retrieval-structural signals match or exceed a 220M-parameter in-domain T5 classifier in most settings with substantially lower variance. Finally, the gains decompose cleanly: the feature framework already outperforms individual heuristics, with ES adding a further 5–7 pp over evaluation-matched random search.

When QDS does not lead. Two regimes constrain the framework. On SciFact at ≥50% budget, the near one-to-one query–document structure makes set-cover near-optimal and a learned combination cannot improve on it. On MSMARCO v1, the in-domain QualT5-zs checkpoint dominates because it has direct access to the training distribution; QDS still beats every purely heuristic baseline and retains practical advantages, including hours rather than days of training, interpretable weights, and no dependence on a 220M-parameter neural backbone. The gap closes on MSMARCO v2 at 120M scale, where QualT5-zs loses its distributional advantage and QDS leads at 3 of 5 budgets.

Limitations and future work. The linear scorer cannot model feature interactions: the MLP variant improves individual cells but averages only +0.6 pp while sacrificing interpretability. A supervised regression or learning-to-rank baseline over the same 13 features is a natural comparison, but requires a differentiable surrogate for nDCG@10, which is precisely the constraint ES removes. Given precomputed features, reoptimisation on shifting workloads is inexpensive: ES takes minutes on BEIR-scale corpora and a few hours at web scale, enabling periodic retraining on fresh query logs and supporting deployment in continuously evolving production environments. Natural extensions include multi-stage retrieval pipelines where document selection interacts with reranking, and finer-grained budgeting that allocates different retention targets to head, torso, and tail query classes rather than a single global budget.

These results suggest that corpus pruning is fundamentally a retrieval-structure problem rather than a document-quality problem, opening directions for workload-aware budgeting in dynamic retrieval systems.

GenAI Usage Disclosure

In compliance with CIKM 2026 and the ACM Policy on Authorship, we disclose the use of Claude (Anthropic) during this work. *Code*: assistance with utility scripts, plotting, and refactoring; all code was reviewed and integrated by the authors. *Writing*: editorial suggestions on phrasing and structure of an author-written manuscript. *Data and experiments*: no GenAI use; all datasets, queries, qrels, and results are from the public benchmarks cited in Section 4 and the authors' own code. The authors take full responsibility for the content of this paper.

References

[1] Antonio Acquavia, Craig Macdonald, and Nicola Tonellotto. 2023. Static Pruning for Multi-Representation Dense Retrieval. In *Proceedings of the ACM Symposium on Document Engineering 2023* (Limerick, Ireland) *(DocEng '23)*. Association for Computing Machinery, New York, NY, USA, Article 7, 10 pages. doi:10.1145/3573128.3604896

[2] Ismail S. Altingovde, Rifat Ozcan, and Özgür Ulusoy. 2012. Static index pruning in web search engines: Combining term and document popularities with query views. *ACM Trans. Inf. Syst.* 30, 1, Article 2 (March 2012), 28 pages. doi:10.1145/2094072.2094074

[3] Leif Azzopardi and Vishwa Vinay. 2008. Retrievability: an evaluation measure for higher order information access tasks. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management* (Napa Valley, California, USA) *(CIKM '08)*. Association for Computing Machinery, New York, NY, USA, 561–570. doi:10.1145/1458082.1458157

[4] Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. 2018. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. arXiv:1611.09268 [cs.CL] https://arxiv.org/abs/1611.09268

[5] Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. 2018. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. (2018). arXiv:1611.09268 [cs.CL] https://arxiv.org/abs/1611.09268

[6] Christopher JC Burges. 2010. From ranknet to lambdarank to lambdamart: An overview. *Learning* 11, 23-581 (2010), 81.

[7] Stefan Büttcher and Charles L. A. Clarke. 2006. A document-centric approach to static index pruning in text retrieval systems. In *Proceedings of the 15th ACM International Conference on Information and Knowledge Management* (Arlington, Virginia, USA) *(CIKM '06)*. Association for Computing Machinery, New York, NY, USA, 182–189. doi:10.1145/1183614.1183644

[8] Jaime Carbonell and Jade Goldstein. 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Melbourne, Australia) *(SIGIR '98)*. Association for Computing Machinery, New York, NY, USA, 335–336. doi:10.1145/290941.291025

[9] David Carmel, Doron Cohen, Ronald Fagin, Eitan Farchi, Michael Herscovici, Yoelle S. Maarek, and Aya Soffer. 2001. Static index pruning for information retrieval systems. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (New Orleans, Louisiana, USA) *(SIGIR '01)*. Association for Computing Machinery, New York, NY, USA, 43–50. doi:10.1145/383952.383958

[10] Xuejun Chang, Zaiqiao Meng, and Debasis Ganguly. 2025. T-Retrievability: A Topic-Focused Approach to Measure Fair Document Exposure in Information Retrieval. *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*. https://api.semanticscholar.org/CorpusID:280985210

[11] Xuejun Chang, Debabrata Mishra, Craig Macdonald, and Sean MacAvaney. 2024. Neural Passage Quality Estimation for Static Pruning. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Washington DC, USA) *(SIGIR '24)*. Association for Computing Machinery, New York, NY, USA, 174–185. doi:10.1145/3626772.3657765

[12] Krzysztof Choromanski, Aldo Pacchiano, Jack Parker-Holder, Yunhao Tang, Deepali Jain, Yuxiang Yang, Atil Iscen, Jasmine Hsu, and Vikas Sindhwani. 2019. Provably Robust Blackbox Optimization for Reinforcement Learning. arXiv:1903.02993 [cs.LG] https://arxiv.org/abs/1903.02993

[13] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, Jimmy Lin, Ellen M. Voorhees, and Ian Soboroff. 2025. Overview of the TREC 2022 deep learning track. arXiv:2507.10865 [cs.IR] https://arxiv.org/abs/2507.10865

[14] Konstantinos Drossos, Samuel Lipping, and Tuomas Virtanen. 2019. Clotho: An Audio Captioning Dataset. arXiv:1910.09387 [cs.SD] https://arxiv.org/abs/1910.09387

[15] Guglielmo Faggioli, Nicola Ferro, Raffaele Perego, and Nicola Tonellotto. 2024. Dimension Importance Estimation for Dense Information Retrieval. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Washington DC, USA) *(SIGIR '24)*. Association for Computing Machinery, New York, NY, USA, 1318–1328. doi:10.1145/3626772.3657691

[16] Uriel Feige. 1998. A threshold of ln n for approximating set cover. *J. ACM* 45, 4 (July 1998), 634–652. doi:10.1145/285055.285059

[17] Chengqian Gao, William de Vazelhes, Hualin Zhang, Bin Gu, and Zhiqiang Xu. 2024. Hard-Thresholding Meets Evolution Strategies in Reinforcement Learning. In *Proceedings of the 33rd International Joint Conference on Artificial Intelligence (IJCAI)*. 3989–3997.

[18] Steven Garcia. 2024. *Search engine optimisation using past queries*. Ph. D. Dissertation. RMIT University.

[19] Sebastian Hofstätter, Sheng-Chieh Lin, Jheng-Hong Yang, Jimmy Lin, and Allan Hanbury. 2021. Efficiently Teaching an Effective Dense Retriever with Balanced Topic Aware Sampling. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Virtual Event, Canada) *(SIGIR '21)*. Association for Computing Machinery, New York, NY, USA, 113–122. doi:10.1145/3404835.3462891

[20] Osman Ali Sadek Ibrahim and Dario Landa-Silva. 2017. ES-Rank: evolution strategy learning to rank approach. In *Proceedings of the Symposium on Applied Computing* (Marrakech, Morocco) *(SAC '17)*. Association for Computing Machinery, New York, NY, USA, 944–950. doi:10.1145/3019612.3019696

[21] Herve Jégou, Matthijs Douze, and Cordelia Schmid. 2011. Product Quantization for Nearest Neighbor Search. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33, 1 (2011), 117–128. doi:10.1109/TPAMI.2010.57

[22] 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)*, Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (Eds.). Association for Computational Linguistics, Online, 6769–6781. doi:10.18653/v1/2020.emnlp-main.550

[23] Aditya Kusupati, Gantavya Bhatt, Aniket Rege, Matthew Wallingford, Aditya Sinha, Vivek Ramanujan, William Howard-Snyder, Kaifeng Chen, Sham Kakade, Prateek Jain, and Ali Farhadi. 2024. Matryoshka Representation Learning. arXiv:2205.13147 [cs.LG] https://arxiv.org/abs/2205.13147

[24] Hoang Thanh Lam, Raffaele Perego, and Fabrizio Silvestri. 2010. On Using Query Logs for Static Index Pruning. In *Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology - Volume 01 (WI-IAT '10)*. IEEE Computer Society, USA, 167–170. doi:10.1109/WI-IAT.2010.139

[25] Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. 2021. Pyserini: A Python Toolkit for Reproducible Information Retrieval Research with Sparse and Dense Representations. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Virtual Event, Canada) *(SIGIR '21)*. Association for Computing Machinery, New York, NY, USA, 2356–2362. doi:10.1145/3404835.3463238

[26] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. 2015. Microsoft COCO: Common Objects in Context. arXiv:1405.0312 [cs.CV] https://arxiv.org/abs/1405.0312

[27] Yuanguo Lin, Fan Lin, Guorong Cai, Hong Chen, Lixin Zou, and Pengcheng Wu. 2024. Evolutionary Reinforcement Learning: A Systematic Review and Future Directions. (2024). arXiv:2402.13296 [cs.NE] https://arxiv.org/abs/2402.13296

[28] Yiming Peng, Gang Chen, Mengjie Zhang, and Bing Xue. 2024. Proximal evolutionary strategy: improving deep reinforcement learning through evolutionary policy optimization. *Memetic Computing* 16, 3 (2024), 445–466.

[29] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning Transferable Visual Models From Natural Language Supervision. arXiv:2103.00020 [cs.CV] https://arxiv.org/abs/2103.00020

[30] Stephen Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Found. Trends Inf. Retr.* 3, 4 (April 2009), 333–389. doi:10.1561/1500000019

[31] Tim Salimans, Jonathan Ho, Xi Chen, Szymon Sidor, and Ilya Sutskever. 2017. Evolution Strategies as a Scalable Alternative to Reinforcement Learning. (2017). arXiv:1703.03864 [stat.ML] https://arxiv.org/abs/1703.03864

[32] Rodrygo L. T. Santos, Craig Macdonald, and Iadh Ounis. 2015. Search Result Diversification. *Found. Trends Inf. Retr.* 9, 1 (March 2015), 1–90. doi:10.1561/1500000040

[33] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. arXiv:2402.03300 [cs.CL] https://arxiv.org/abs/2402.03300

[34] Federico Siciliano, Francesca Pezzuti, Nicola Tonellotto, and Fabrizio Silvestri. 2024. Static Pruning in Dense Retrieval using Matrix Decomposition. arXiv:2412.09983 [cs.IR] https://arxiv.org/abs/2412.09983

[35] Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A Heterogenous Benchmark for Zero-shot Evaluation of Information Retrieval Models. arXiv:2104.08663 [cs.IR] https://arxiv.org/abs/2104.08663

[36] Nicola Tonellotto and Craig Macdonald. 2021. Query Embedding Pruning for Dense Retrieval. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management* (Virtual Event, Queensland, Australia) *(CIKM '21)*. Association for Computing Machinery, New York, NY, USA, 3453–3457. doi:10.1145/3459637.3482162

[37] Henning Wachsmuth, Shahbaz Syed, and Benno Stein. 2018. Retrieval of the Best Counterargument without Prior Topic Knowledge. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Iryna Gurevych and Yusuke Miyao (Eds.). Association for Computational Linguistics, Melbourne, Australia, 241–251. doi:10.18653/v1/P18-1023

[38] Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. MINILM: deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Proceedings of the 34th International Conference on Neural Information Processing Systems* (Vancouver, BC, Canada) *(NIPS '20)*. Curran Associates Inc., Red Hook, NY, USA, Article 485, 13 pages.

[39] Yusong Wu, Ke Chen, Tianyu Zhang, Yuchen Hui, Marianna Nezhurina, Taylor Berg-Kirkpatrick, and Shlomo Dubnov. 2024. Large-scale Contrastive Language-Audio Pretraining with Feature Fusion and Keyword-to-Caption Augmentation. arXiv:2211.06687 [cs.SD] https://arxiv.org/abs/2211.06687

[40] Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. 2024. C-Pack: Packed Resources For General Chinese Embeddings. (2024), 641–649. doi:10.1145/3626772.3657878

[41] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate Nearest Neighbor Negative Contrastive Learning for Dense Text Retrieval. arXiv:2007.00808 [cs.IR] https://arxiv.org/abs/2007.00808

[42] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. 2025. Qwen3 Technical Report. (2025). arXiv:2505.09388 [cs.CL] https://arxiv.org/abs/2505.09388

[43] 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*, Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (Eds.). Association for Computational Linguistics, Brussels, Belgium, 2369–2380. doi:10.18653/v1/D18-1259

[44] Chao Zhang and Renée J. Miller. 2026. Distribution-Aware Exploration for Adaptive HNSW Search. *Proc. ACM Manag. Data* 4, 1, Article 25 (April 2026), 27 pages. doi:10.1145/3786639

[45] Xiaolan Zhu and Susan Gauch. 2000. Incorporating Quality Metrics in Centralized/Distributed Information Retrieval on the World Wide Web. In *SIGIR*. 288–295. doi:10.1145/345508.345602