More Than Efficiency: Embedding Compression Improves Domain Adaptation in Dense Retrieval

Chunsheng Zuo
Johns Hopkins University
czuo3@jh.edu

Daniel Khashabi
Johns Hopkins University
danielk@jhu.edu

Abstract

Dense retrievers powered by pretrained embeddings are widely used for document retrieval but struggle in specialized domains due to the mismatches between the training and target domain distributions. Domain adaptation typically requires costly annotation and retraining of query-document pairs. In this work, we revisit an overlooked alternative: applying PCA to domain embeddings to derive lower-dimensional representations that preserve domain-relevant features while discarding non-discriminative components. Though traditionally used for efficiency, we demonstrate that this simple embedding compression can effectively improve retrieval performance. Evaluated across 9 retrievers and 14 MTEB datasets, PCA applied solely to query embeddings improves NDCG@10 in 75.4% of model-dataset pairs, offering a simple and lightweight method for domain adaptation.

1 Introduction

With recent advancements in retrieval-augmented generation (Lewis et al., 2020b; Fan et al., 2024), dense retrievers (Karpukhin et al., 2020) have become increasingly prominent. These models produce vector embeddings that encode semantic information from text, enabling effective matching and retrieval of contextually relevant documents. However, dense retrievers pretrained on general, large-scale corpora inherently encode semantic features reflecting the training data distribution (Reimers and Gurevych, 2019). Consequently, when applied to specialized domains such as biomedical or finance, these pretrained retrievers face challenges due to distribution shifts between training data and domain-specific text (Lupart et al., 2023). Such shifts can lead retrievers to overlook critical domain-specific information or fail to capture essential nuances during the retrieval process.

Recent works (Gururangan et al., 2020; Siriwardhana et al., 2023; Hashemi et al., 2023; Li and Gaussier, 2024) have addressed domain adaptation by fine-tuning retrievers on annotated domain-specific datasets, typically comprising query-document pairs. Nonetheless, acquiring high-quality domain-specific annotated data can often be tedious due to annotation costs and data availability constraints.

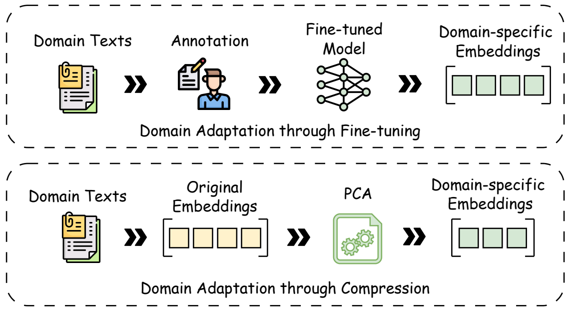

Figure 1: PCA compresses embeddings to become domain-specific, providing a more efficient domain adaptation method than traditional fine-tuning, with an additional advantage of a lower retrieval cost.

In this work, we explore a simpler yet overlooked approach for domain adaptation: dimensionality reduction through Principal Component Analysis (PCA) (Abdi and Williams, 2010). While traditionally employed for dimensionality reduction to speed up retrieval (Ma et al., 2021), we hypothesize that PCA also highlights principal dimensions critical to the domain's semantic space, refocusing retrieval on domain-relevant information without introducing new knowledge. We posit that PCA, by identifying principal components in a domain-specific corpus or query embeddings, implicitly highlights dimensions critical to the domain's semantic space. Retaining only the top principal dimensions thus refocuses the retriever's attention on domain-relevant information.

Our contribution includes:

- We demonstrate that a simple application of PCA to queries alone improves retrieval performance (NDCG@10) in a majority (75.4%)

Proceedings of the First Workshop on Structured Understanding, Retrieval, and Generation in the LLM Era (SURGeLLM 2026), pages 361–377
July 3, 2026 ©2026 Association for Computational Linguistics

of the evaluated model-dataset combinations.

- To the best of our knowledge, we are the first to establish that fitting PCA solely on queries is more effective for domain adaptation than using both queries and documents.

- We investigate the relationship between model domain familiarity and performance gains from PCA, revealing divergent, model-specific adaptation patterns.

- We show that PCA, a zero-cost method, can outperform a state-of-the-art compressive domain adaptation technique (Thakur et al., 2023) that relies on computationally expensive pseudo-label curation and fine-tuning.

2 Related Works

Unsupervised Domain Adaptation with Synthetic Data: Recent works have explored domain adaptation through synthetic data generation and pseudo-labeling. Wang et al. (2021, 2022) leverage denoising auto-encoders and pseudo-relevance labeling to adapt retrievers without annotated data, while Hashemi et al. (2023); Meng et al. (2022); Bonifacio et al. (2022) generate synthetic queries and documents through text generation models. Building on Generative Pseudo Labeling (GPL) (Wang et al., 2022), the work Thakur et al. (2023) apply Joint Product Quantization (Zhan et al., 2021) to a GPL-trained Tas-B model (Hofstätter et al., 2021) to enable compact, efficient retrieval at a modest effectiveness cost. These approaches demonstrate that synthetic data can mitigate domain shift, but they require substantial computation for data generation and fine-tuning. In contrast, our study isolates the underexplored role of PCA as a simple, training-free compression/adaptation technique.

Few-Shot and Zero-Shot Adaptation: Alternative approaches minimize annotation requirements by using large language models (LLMs) for few-shot data generation. Dai et al. (2022) create task-specific retrievers from as few as 8 examples via LLM-generated queries, and Huang et al. (2023) adapt conversational retrievers using only 6 dialogue examples. While effective, these methods depend on the availability of powerful LLMs and carefully crafted prompts. Additionally, Promptriever (Weller et al., 2024) can adapt to a new domain by following task-related instructions. Methods such as Morris and Rush (2024) adapt retrievers by modifying the architecture and training objective to incorporate neighboring-document information, improving out-of-domain retrieval.

Despite avoiding reliance on annotations, these methods require substantial changes to the model's architecture or training process, inapplicable to other already-trained models.

PCA for Retrieval: Prior work has used dimensionality reduction primarily to improve retrieval efficiency (Ma et al., 2021). Raunak et al. (2019) apply PCA to pre-trained embeddings for downstream task performance, while Chen et al. (2020) binarizes PCA-compressed embeddings for image retrieval. However, these methods either optimize for training data characteristics or sacrifice embedding expressivity, limiting their adaptability.

3 Preliminaries and Notation

We set up the framework and notations, starting with defining the retrieval problem (§3.1), followed by matrix projection (§3.2).

3.1 Standard Dense Retrieval

The retrieval problem: Given a collection of documents $D = \{d_j\}_{j=1}^m$ and a set of likely queries $Q = \{q_i\}_{i=1}^n$, the fundamental task is to, given query $q_i \in Q$, retrieve a ranked list of the most relevant documents in D:

$$\text{top-}k_{j\in[1,m]}(\text{sim}(q_i, d_j)), \tag{1}$$

where $\text{sim}(.,.)$ is a similarity metric that quantifies relevance between a query q_i and a document d_j.

Dense retrieval: In the special case of dense retrieval, a pre-trained neural network, which we denote as the ENC, maps both queries and documents into shared high-dimensional embeddings:

$$\mathbf{q}_i \leftarrow \text{ENC}(q_i) \in \mathbb{R}^d,\ \mathbf{d}_j \leftarrow \text{ENC}(d_j) \in \mathbb{R}^d.$$

Given these dense representations, the similarity between a query q_i and a document d_j is computed via the cosine similarity between their d-dimensional representation:

$$\text{sim}(q_i, d_j) := \frac{\mathbf{q}_i \cdot \mathbf{d}_j}{\|\mathbf{q}_i\|\,\|\mathbf{d}_j\|}, \tag{2}$$

which is used to rank and retrieve documents for each query based on this similarity score. Our work builds upon this foundation by introducing a dimensionality reduction step for domain adaptation after the initial encoding.

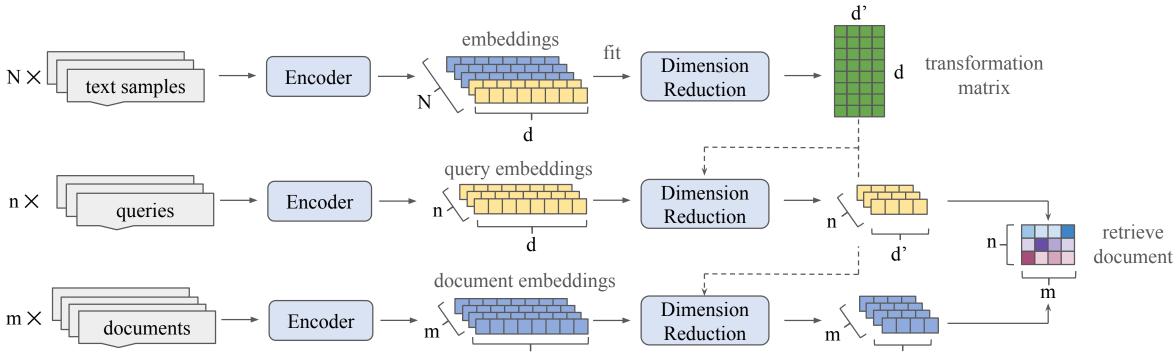

Figure 2: Pipeline for adapting embeddings to a test time domain using PCA (§3.2). The encoder in the top branch processes text samples (queries, documents, or a mixture) from the test domain. These samples are used by the PCA algorithm to learn a projection matrix of the top d' principal components. This matrix then transforms both the test domain's query and document embeddings. Finally, a similarity match is performed on the transformed embeddings to retrieve the most relevant documents.

3.2 Matrix Projection with PCA

PCA is a widely used framework for projecting high-dimensional data to a lower-dimensional space. Let $\mathbf{X} \in \mathbb{R}^{k \times d}$ represent the matrix of k items (e.g., documents), each with d-dimensional embeddings. The first step is mean-centering. Let $\tilde{\mathbf{X}} = \mathbf{X} - \mathbf{1}_k \boldsymbol{\mu}^\top$ be the mean-centered version, where $\boldsymbol{\mu} = \frac{1}{k}\mathbf{X}^\top \mathbf{1}_k$ is the column-wise mean vector and $\mathbf{1}_k$ is a k-dimensional vector of ones.

To reduce dimensionality from $\mathbb{R}^{k \times d}$ to $\mathbb{R}^{k \times d'}$, PCA finds an orthogonal projection matrix $\mathbf{W} \in \mathbb{R}^{d \times d'}$ that maximizes variance in the projected data:

$$\mathbf{W}^* = \text{argmax}_{\mathbf{W}^\top \mathbf{W} = I_{d'}} \mathbf{W}^\top \tilde{\mathbf{X}}^\top \tilde{\mathbf{X}} \mathbf{W}, \quad (3)$$

where $I_{d'}$ is the $d' \times d'$ identity matrix.

Solving this objective: The solution $\mathbf{W}^*$ consists of the top d' *eigenvectors* of the sample covariance matrix $\frac{1}{k-1}\tilde{\mathbf{X}}^\top \tilde{\mathbf{X}}$, corresponding to the directions of greatest variance. The associated *eigenvalues* quantify how much variance each principal axis explains. This solution can also be obtained directly via the Singular Value Decomposition (SVD) of the mean-centered matrix $\tilde{\mathbf{X}}$.

4 Method: Unsupervised Representation Compression for Retrieval

We detail our approach for adapting pre-trained dense retrievers to a target domain using dimensionality reduction.

Retrieval on down-projected embeddings: The goal is to project the high-dimensional embeddings into a lower-dimensional subspace $\mathbb{R}^{d'}$ (where $d' < d$) that effectively captures the semantic structure of the target domain, by discarding information that does not allow distinguishing the target domain. Formally, let $\mathbf{W}$ be a projection matrix used to linearly transform both query and document embeddings into the lower-dimensional space:

$$\mathbf{q}'_i \leftarrow \mathbf{q}_i \mathbf{W}, \qquad \mathbf{d}'_j \leftarrow \mathbf{d}_j \mathbf{W}.$$

These down-projections effectively re-align the original embedding features, emphasizing semantic components aligned with the target domain. Note that none of this involves adapting the parameters of the pre-trained `Encoder`.

Using the down-projected embeddings, retrieval is carried out in the reduced space as defined in Eq. 1 and 2. We now describe how the projection matrix $\mathbf{W}$ is derived.

Down-projection with PCA: At the core of our approach is learning a projection matrix $\mathbf{W} \in \mathbb{R}^{d \times d'}$ that reduces the dimensionality of embeddings obtained from the target domain. This matrix is obtained by applying Principal Component Analysis (PCA) to embeddings produced by a fixed, pre-trained dense retriever (see Fig. 2). Specifically, we explore two strategies for fitting the PCA:

(1) *Query Compression:* The PCA model is computed exclusively on the set of query embeddings $\{\mathbf{q}_i\}_{i=1}^n$. This approach aims to find a subspace that maximizes the variance observed within the query distribution, potentially highlighting dimensions crucial for distinguishing query intents in the target domain.

(2) *Query+Document Compression:* The PCA model is computed on the union of query and

document embeddings, $\{\mathbf{q}_i\}_{i=1}^{n} \cup \{\mathbf{d}_j\}_{j=1}^{m}$. This captures the variance across the entire target corpus to find a subspace shared by both query and document semantics.

See Alg. 1 for the details of the construction.

Algorithm 1: PCA Domain Adaptation

Input: `Encoder` (pretrained retriever), query set $Q = \{q_i\}_{i=1}^{n}$, document set $D = \{d_j\}_{j=1}^{m}$, target dim $d' < d$
Output: Ranked document list for each $q_i \in Q$

Encode queries: $\mathbf{q}_i \leftarrow \texttt{Encoder}(q_i)$ for all $q_i \in Q$
Encode docs: $\mathbf{d}_j \leftarrow \texttt{Encoder}(d_j)$ for all $d_j \in D$
Fit PCA on $\{\mathbf{q}_i\}$ or $\{\mathbf{q}_i\} \cup \{\mathbf{d}_j\}$ to obtain mean $\boldsymbol{\mu}$ and projection matrix $\mathbf{W} \in \mathbb{R}^{d \times d'}$
Project: $\mathbf{x}' \leftarrow (\mathbf{x} - \boldsymbol{\mu})\mathbf{W}$ for all $\mathbf{x} \in \{\mathbf{q}_i\} \cup \{\mathbf{d}_j\}$
foreach $q_i \in Q$ **do**
  foreach $d_j \in D$ **do**
    Compute similarity: $s_{ij} \leftarrow \frac{\mathbf{q}'_i \cdot \mathbf{d}'_j}{\|\mathbf{q}'_i\| \|\mathbf{d}'_j\|}$
  Rank D by s_{ij} in descending order
return ranked document lists for all q_i

Assumptions: Our approach assumes access to a corpus of documents and queries from the target domain. Importantly, it operates solely on raw queries and documents, *without* relying on labeled query-document relevance pairs typically required for supervised retrieval. This is a practical assumption in many settings where unlabeled data is abundant.

4.1 Insights Behind Embedding Projection

IR representations are low-rank: As noted in §3.2, PCA identifies a low-dimensional subspace whose eigenvectors capture the directions of greatest variance in the target-domain embeddings. The corresponding eigenvalue for each principal axis represents its importance. We noticed that, similar to the examples shown in Fig. 3, the eigenvalue spectra for the embeddings of a dataset typically exhibit a power-law-like decay (See Appendix A), facilitating lower-dimensional representation.

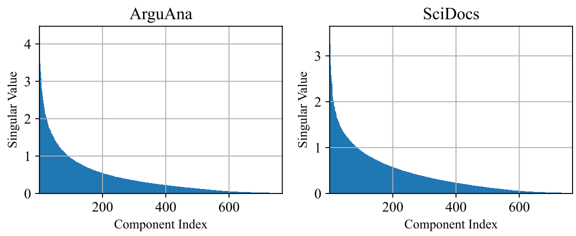

Figure 3: The distribution of the eigenvalues for the principal components after applying PCA to fit the queries' embeddings of the Sentence-T5 model on each dataset (ArguAna and SciDocs).

The projection helps domain adaptation by preserving salient information for the target: The core hypothesis is that these high-variance directions preferentially capture the most salient semantic variations specific to the *target* domain. Projecting onto the subspace defined by these directions filters out low-variance components—often associated with *source*-domain artifacts or noise—thereby improving focus on the *target* domain.

This approach relies on three key premises:

1. The pre-trained encoder, even without fine-tuning, captures sufficient domain-relevant information within its embeddings.
2. The primary semantic variations characteristic of the *target* domain manifest as directions of high variance in the embedding space.
3. Variance from the *source* domain or noise often lies orthogonal to the *target* domain's principal components and is attenuated by projection.

5 Experimental Setup

Variations: We evaluate three PCA compression strategies. **No Compression (Baseline):** Use raw model embeddings without dimensionality reduction. **Query Compression:** Fit PCA on domain-specific query embeddings and apply the projection to both queries and documents. **Query+Document Compression:** Fit PCA on the combined query and document embeddings to capture broader domain variance. The retention ratio r is defined as:

$$r = \lfloor d'/d \rfloor, \tag{4}$$

where d is the original embedding dimension and d' is the number of retained principal components. Unless otherwise noted, we use $r = 0.9$ in our experiments (§6.1) and also explore the trade-offs across retention ratios from 0.1 to 1.0 in increments of 0.1.

Datasets: We assess PCA-based adaptation on 14 diverse datasets from the MTEB benchmark (Muennighoff et al., 2023), comparing against performance with uncompressed embeddings (Appendix I). We also include 11 additional MTEB datasets (Appendix E) where the number of queries for each is smaller than the target PCA dimension, leading to lower-than-intended retention ratios.

These datasets span multiple domains (e.g., software engineering, biomedical, finance, and scientific literature, news), and even multiple languages (English, Danish, and German), allowing a com-

Datasets↓	Query Compression									Query+Document Compression								
	MiniLM	GTE	Instr.	Dis-Ro	MP-QA	MP-All	Sent-T5	BGE	SFR	MiniLM	GTE	Instr.	Dis-Ro	MP-QA	MP-All	Sent-T5	BGE	SFR
Code	1.2	2.3	-1.0	6.0	1.1	1.3	38.7	4.0	-0.2	-0.7	1.0	-1.8	0.7	-1.0	-1.3	18.8	-1.3	-0.6
Apps	-4.1	1.1	-14.5	-16.2	1.1	-11.1	18.0	-16.1	-0.2	-21.1	-1.5	-24.8	-52.8	-12.4	-22.1	5.1	-36.8	-1.2
SciDocs	-0.2	18.5	3.0	-0.5	-1.0	3.3	3.8	4.8	2.1	-0.4	13.3	2.5	0.5	-0.3	2.4	-5.2	-7.9	1.9
MedQA	2.1	2.0	2.1	2.7	2.2	1.9	6.9	0.4	0.5	1.9	1.4	1.6	2.2	1.3	0.9	5.6	-0.4	0.5
ArguAna	1.2	-2.2	3.0	0.6	0.3	-0.3	11.1	2.2	-1.3	0.4	-2.5	1.6	-0.4	-1.1	-1.4	7.1	0.4	-2.0
StackOverflow	0.1	-0.1	-1.1	0.4	-0.1	1.2	2.9	0.4	-0.9	0.1	0.0	-1.1	-0.8	-0.1	0.4	2.9	-0.1	-1.1
TV2Nord	40.2	4.4	2.8	17.1	17.7	7.4	1.2	0.3	5.5	24.9	2.7	2.6	11.7	4.2	4.0	-0.6	-0.2	2.7
GerDa	6.2	3.5	15.7	-1.4	7.4	7.0	3.2	1.8	7.8	-1.7	-10.1	2.0	-10.8	-1.6	-0.6	-7.9	-14.2	-13.4
ARC	2.5	1.4	12.9	2.7	3.0	2.0	11.1	7.3	4.4	3.7	1.7	13.1	2.3	3.8	1.9	9.4	8.2	3.0
FeedbackQA	-1.9	1.0	-3.7	0.2	-2.2	-3.2	-2.4	-0.4	-0.2	-2.7	0.5	-6.0	-6.9	-5.3	-3.5	-4.6	-1.2	-1.2
FaithDial	0.3	5.1	4.0	0.8	0.8	1.8	4.2	7.9	4.8	1.3	4.8	2.3	-0.2	1.3	0.8	3.1	7.9	4.6
MLQA	0.5	0.6	2.5	0.8	1.0	0.4	-0.4	0.2	1.8	0.2	0.6	2.5	0.4	0.8	0.3	-1.6	0.3	1.7
NarrativeQA	11.0	41.5	7.1	15.3	13.1	5.8	20.5	1.6	2.6	7.1	37.0	4.7	11.6	8.5	1.5	17.4	1.1	1.6
SpartQA	620.0	149.5	343.0	508.4	1954.6	511.0	119.7	192.5	849.2	626.7	173.4	717.7	463.1	1509.1	632.3	248.1	209.0	829.7
Summary	11/14	12/14	10/14	10/14	11/14	10/14	12/14	10/14	9/14	9/14	8/14	8/14	7/14	8/14	8/14	9/14	6/14	8/14

Table 1: NDCG@10 improvement (%) comparison between Query Compression (left) and Query+Document Compression (right). Positive gains are highlighted green for Query Compression and blue for Query+Document Compression, negative values are red. The "Summary" row indicates the number of datasets with improvements, per model. The table shows that both Query Compression and Query+Document Compression improve retrieval quality. However, **Query Compression yields more consistent gains, improving 95 out of 126 data–model pairs (75.4%)**, compared to 71 out of 126 (56.3%) for Query+Document Compression. More detailed results with raw performance values can be found in Appendix F.

prehensive evaluation of PCA's effectiveness in handling various distribution shifts.

Models: We evaluate 9 popular pretrained dense retrieval models with fewer than 2B parameters. Our selection includes five models from the Sentence Transformers library (`Dis-Ro`, `MP-QA`, `MiniLM`, `MP-All`, `Sent-T5`) and four other widely used models: `BGE`, `GTE`, `SFR`, and `Instr`. A complete list of the models, their abbreviations, and corresponding references is provided in Appendix J.

Metrics and evaluation: We primarily use `NDCG@10` to evaluate retrieval performance. We have also added results for Precision@10 and Recall@10 in Appendix C.

Overall, each data-model pair is evaluated under three configurations (baseline, Query Compression, and Query+Document Compression). With 9 models and 14 + 11 datasets, this results in a total of 675 retrieval runs. All experiments are conducted with an RTX 4090 GPU for around 36 hours.

Dataset	# of Queries	# of Docs	Success Rate
MedQA	2048	2048	9/9
SpartQA	3594	1592	9/9
FaithDial	2042	3539	9/9
NarrativeQA	10557	355	9/9
ARC	1172	9350	9/9
TV2Nord	2048	2048	9/9
MLQA	11582	9916	8/9
GerDa	12234	9969	8/9
Code	14918	280310	7/9
ArguAna	1406	8674	6/9
SciDocs	1000	25657	6/9
StackOverflow	1994	19931	5/9
Apps	3765	8765	3/9
FeedbackQA	1992	2364	2/9

Table 2: Per-dataset summary statistics. For each dataset, we report its size (# of queries and documents) and the success rate of 90% Query Compression—defined as the proportion of embedder representations that see improved performance after compression. As shown in the success rate column, **some datasets consistently benefit from query-only compression across all or many choices of representations**.

6 Empirical Results

6.1 Main Findings

Query Compression Generally Improves Performance: Table 1 presents the NDCG@10 improvements obtained from applying 90% Query Compression across various dataset-model combinations. Query-only compression consistently demonstrates effectiveness across diverse domains, with improvements observed in 75.4% of model-dataset pairings. All models, except for SFR, achieve improvements in 10 to 11 datasets. Notably, GTE and Sent-T5 show positive gains across 12 out of 14 datasets, making them the most robust models for Query Compression. Most degradations are within 4%. In contrast, several positive outliers are substantial: TV2Nord + MiniLM, Nar-

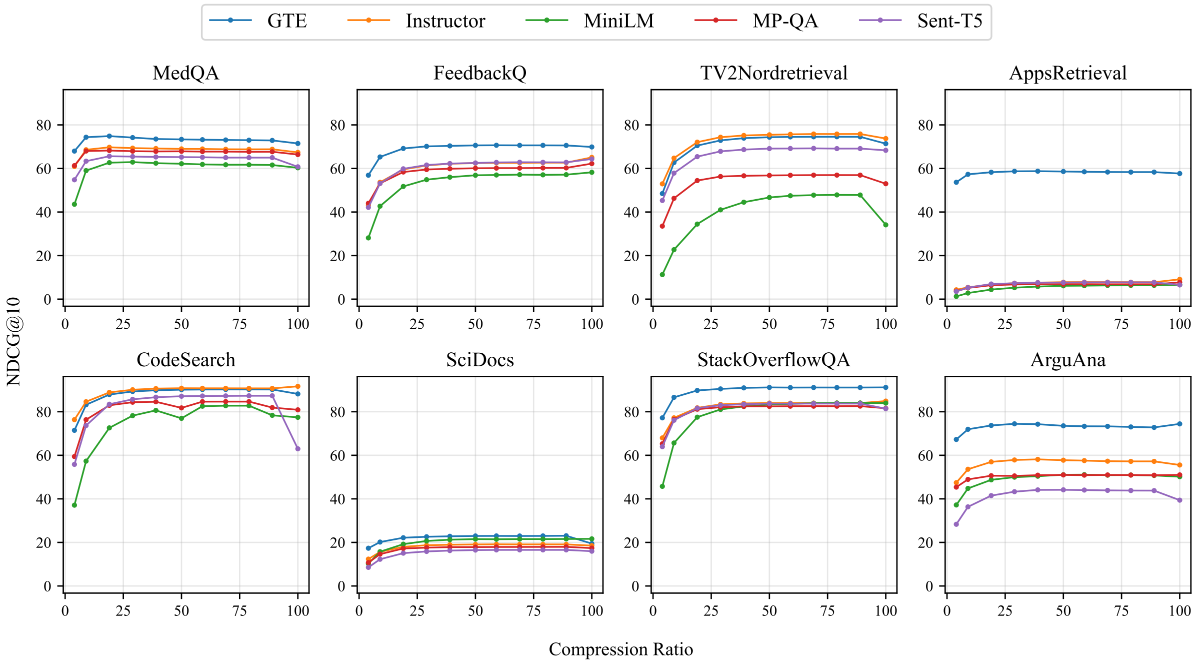

Figure 4: NDCG@10 at various retention ratios ranging from 0.1 to 1, with step size of 0.1 and an additional inclusion of a very extreme 0.05. **While the optimal Query retention ratio is dataset- and model-dependent, in most cases where Query Compression dominates the original embeddings, the dominance can continue even until the retention ratio is around 50%.**

rativeQA + GTE, and Code + Sent-T5 each achieve gains of over 40%.

Table 2 summarizes the success rate of Query Compression on the 14 datasets, where 12 of them have the majority of models improved, and 6 of them have all models improved. Whether a dataset can be successfully adapted through compression is not dependent on the number of queries it contains. Overall, these results highlight that Query Compression generally allows the model to adapt to a new domain with minimal loss or even considerable performance gains.

Query+Document Compression underperforms Query Compression: As a general observation in Table 1, while Query Compression achieves improvements on 75.4% of data-model combinations, the percentage drops to 56.3% for Query+Document Compression. On the Apps dataset, for example, nearly all models experience a substantial performance drop with Query+Document Compression, exemplified by a drastic fall of up to -52.8% for Dis-RoBERTa. A similar pattern is visible on the GerDa dataset, where Query Compression shows mostly gains, but Query+Document Compression results in significant performance loss for almost every model. This indicates that queries alone are better at capturing domain-specific user information needs, which aligns the PCA projection more precisely with the search task. In contrast, high-variance directions in the full document corpus may reflect broad topical or stylistic variation rather than the relevance distinctions expressed by the queries, thereby diluting this effect. Despite a few dataset-model combinations that benefit more from Query+Document compression, the overall evidence shows that query-only compression remains the safer, more consistently effective strategy.

Hierarchical Query Structures Amplify PCA Gains: While Table 1 shows that PCA provides widespread benefits, the magnitude of these gains varies significantly across datasets. We argue that this variance is strongly linked to the inherent structure of the domain; specifically, domains with clear hierarchical or categorical query patterns benefit the most from PCA. This effect is exemplified by SpartQA, a dataset organized by a logical structure of attributes like shapes, colors, and spatial relations. Its queries possess a highly regular, almost combinatorial nature, which helps explain the outsized performance improvements observed across all models. Similarly, for MedQA, where models consistently achieve positive gains, the queries are classifiable by a certain medical hierarchy (e.g., diseases, symptoms, treatments). In such structured domains, the principal components identified by

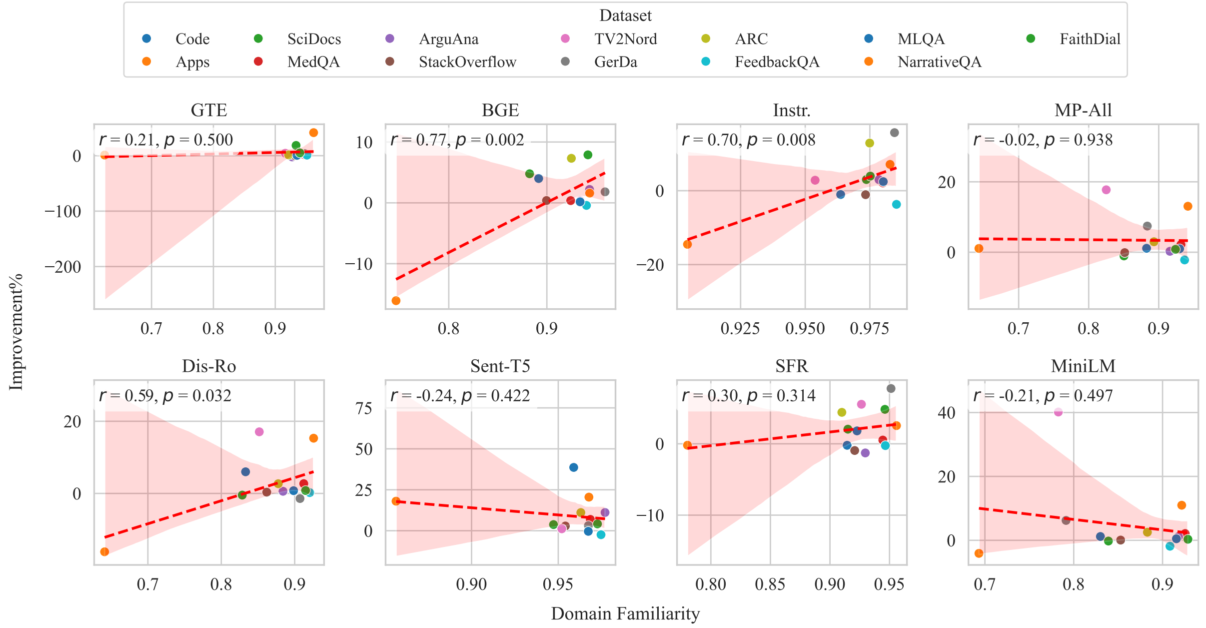

Figure 5: Correlation between domain familiarity (Equation 6) and performance improvement after 90% Query Compression. **For BGE, Instructor, and Dis-Ro, retrieval performance after compression benefits from higher domain familiarity.**

PCA are more likely to correspond directly to these meaningful semantic axes. The projection, therefore, acts as a targeted feature selection process, magnifying the benefits of compression.

6.2 Further Analysis

Performance at various Retention Ratios: We further investigate the trade-off between performance and retention ratio. Figure 4 shows the NDCG@10 at different Query Retention ratios for 8 of the 14 datasets.

In general, a moderate retention ratio optimizes performance before information loss leads to decline. On CodeSearch, MiniLM peaked at 82.8 NDCG@10 with 80% of dimensions, up from 77.4, while Sent-T5 saw a dramatic rise from 63.0 to 87.3 with 90% of dimensions. This suggests an optimal balance between conciseness and informational integrity.

Notably, certain datasets are more tolerant of compression. On ArguAna and MedQA, the performance remains strong down to a 40% retention ratio. GTE even peaks in MedQA at 10% compression, equivalent to embeddings with 76 dimensions. We attribute this to the quality of MedQA queries that consist of systematic questions, providing valuable guidance for domain-specific projection. Furthermore, rather than indicating that a few components are essential, being robust to compression could mean that the remaining principal components are sufficiently representative to distinguish between queries/documents within the dataset.

Does PCA Compression Adapt Models Better to Niche Domains than Familiar Ones? To investigate whether PCA compression enhances model adaptation to niche domains further, we analyzed the relationship between a model's intrinsic familiarity with a target domain and its post-PCA performance gain. We quantify domain familiarity through **paraphrasing robustness**, measuring how consistently a model represents semantic equivalence under textual variations. Formally, given a text snippet t_i and its paraphrases $\{t_i^1, \ldots, t_i^n\}$, an encoder E's familiarity to t_i is defined as:

$$\mathrm{TF}_n(E, t_i) = \frac{1}{n} \sum_{j=1}^{n} \mathrm{sim}\left(E(t_i), E(t_i^j)\right), \quad (5)$$

where sim denotes cosine similarity. The domain familiarity for dataset $D = \{t_1, \ldots, t_m\}$ with respect to E is then:

$$\mathrm{DF}_m(E, D) = \frac{1}{m} \sum_{i=1}^{m} \mathrm{TF}_n(E, t_i). \quad (6)$$

Higher DF indicates stronger familiarity, as the model produces stable embeddings for paraphrased content within the domain. We randomly sampled 10 queries from each dataset, each producing 3 paraphrases (m=10, n=3).[1] Fig. 5 shows the FD

[1]Despite the small size, the variance was acceptably low.

against the post-PCA (90% compression) performance gain on each dataset [2] for 8 of the models.

The results reveal varied trends. Sent-T5, MiniLM, and MP-All exhibit weak negative to negligible correlations, all statistically insignificant. Conversely, Instructor (Instr.) and BGE demonstrate strong, statistically significant positive correlations. Distilled Roberta (Dis-Ro) also shows a statistically significant, moderate positive correlation. Finally, GTE and SFR indicate weak positive correlations that are not statistically significant.

The interpretation of these mixed results remains complex. On one hand, lower domain familiarity (and thus lower DF) might signal an opportunity for PCA to improve embeddings by removing task-irrelevant variance that dominates in unfamiliar contexts; this could align with the negative correlations observed in Sent-T5 and MiniLM. On the other hand, strong positive correlations, as seen with Instructor, BGE, and Distilled Roberta, suggest that PCA might be more effective when the model already has a foundational understanding (higher familiarity) of the domain, possibly by refining already relevant features. The statistically insignificant results for several models, and the presence of both positive and negative trends, suggest that the relationship between domain familiarity and PCA's impact is model-dependent and potentially influenced by other factors not captured by DF alone.

6.3 Comparison with Other Domain-Adaptation Methods

Our main experiments isolate PCA compression as a training-free adaptation mechanism: it requires neither synthetic data generation nor model fine-tuning. This isolation keeps the 9-model, 14-dataset study focused on the effect of the projection itself, rather than on the many design choices introduced by heavier adaptation pipelines. To connect this analysis with stronger adaptation methods, we additionally compare PCA with fine-tuning methods on the subset of datasets shared by both studies.

Specifically, we compare with Injecting Domain Adaptation (IDA) by Thakur et al. (2023). IDA augments a GPL-adapted Tas-B encoder with deep hashing schemes—Joint Product Quantization (JPQ) (Zhan et al., 2021) and Binary Passage Retrieval (BPR) (Yamada et al., 2021)— to obtain a compact index. We evaluate on the five datasets used by both our work and IDA. Table 3 reports relative improvements over the base Tas-B encoder; the numbers for GPL, GPL + JPQ, and GPL + BPR are taken directly from IDA.

Dataset	PCA	GPL	GPL + JPQ	GPL + BPR
SciDocs	12.1	13.4	4.0	2.0
ArguAna	10.3	29.8	3.3	-17.4
FiQA	14.3	14.6	-1.6	3.0
NFCorpus	1.9	8.1	1.5	-6.2
SciFact	3.2	4.8	1.8	-3.1

Table 3: Relative improvement (%) over the original Tas-B encoder. PCA matches GPL on SciDocs and FiQA, and outperforms both GPL + JPQ and GPL + BPR on all 5 datasets.

PCA matches GPL on SciDocs and FiQA, while GPL remains stronger on ArguAna, NFCorpus, and SciFact. PCA also outperforms both compact IDA variants, GPL + JPQ and GPL + BPR, on all five shared datasets. These results suggest that PCA is not a replacement for full synthetic-data fine-tuning when compute is available, but it is a strong training-free baseline and a practical first step before practitioners pay the cost of data generation, pseudo-label curation, and fine-tuning.

7 Conclusion

Our analysis demonstrates that unsupervised PCA-based compression is an effective and training-free mechanism for domain adaptation in dense retrieval systems. It is proven to be competitive even against heavyweight pipelines (Section 6.3). By identifying principal axes from the variance patterns of the target domain queries, PCA achieves consistent performance gains across diverse models and datasets, particularly through query-only compression that outperforms Query+Document Compression (Section 6.1). Our findings indicate that moderate dimensionality reduction (preserving 50-90% of components) frequently enhances retrieval performance (Figure 4). Notably, these gains persist even when models exhibit varying degrees of pre-existing domain familiarity (Figure 5), underscoring PCA's general utility. This paradigm is especially valuable when labeled domain data is scarce, though practitioners should validate retention ratios for their specific use case. Future work could explore integrating PCA with synthetic data generation or nonlinear manifold learning.

[2]We do not include SpartQA in this analysis because its performance gain is an outlier.

8 Limitations

While our approach demonstrates promising results, several limitations merit consideration. First, the method assumes pre-trained encoders already capture sufficient domain-relevant information–for highly specialized domains with unique terminology, PCA's ability to enhance performance is constrained by the original model's knowledge boundaries. Second, though unsupervised, our technique still requires an adequate amount of unlabeled target domain samples for reliable covariance estimation; insufficient queries may lead to unstable component extraction. Third, the optimal retention ratio proves model- and dataset-dependent, necessitating empirical validation rather than offering universal prescriptive guidelines. In practice, we use $r = 0.9$ as a conservative default and recommend sweeping a small set of retention ratios when validation data is available, falling back to the original embeddings when compression hurts performance. Developing automatic criteria for choosing the optimal retention ratio remains an important direction for future work.

References

Hervé Abdi and Lynne J Williams. 2010. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459.

Asma Ben Abacha and Dina Demner-Fushman. 2019. A question-entailment approach to question answering. *arXiv preprint arXiv:1901.08079*.

Paheli Bhattacharya, Kripabandhu Ghosh, Saptarshi Ghosh, Arindam Pal, Parth Mehta, Arnab Bhattacharya, and Prasenjit Majumder. 2019. Fire 2019 aila track: Artificial intelligence for legal assistance. In *Proceedings of the 11th Annual Meeting of the Forum for Information Retrieval Evaluation*, FIRE '19, page 4–6, New York, NY, USA. Association for Computing Machinery.

Luiz Bonifacio, Hugo Abonizio, Marzieh Fadaee, and Rodrigo Nogueira. 2022. Inpars: Unsupervised dataset generation for information retrieval. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '22, page 2387–2392, New York, NY, USA. Association for Computing Machinery.

Vera Boteva, Demian Gholipour, Artem Sokolov, and Stefan Riezler. 2016. A full-text learning to rank dataset for medical information retrieval. In *Advances in Information Retrieval: 38th European Conference on IR Research, ECIR 2016, Padua, Italy, March 20–23, 2016. Proceedings 38*, pages 716–722. Springer.

Jianlyu Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. M3-embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 2318–2335, Bangkok, Thailand. Association for Computational Linguistics.

Rongyu Chen, Lili Pan, Yan Zhou, and Qianhui Lei. 2020. Image retrieval based on deep feature extraction and reduction with improved cnn and pca. *Journal of Information Hiding and Privacy Protection*, 2(2):67.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge.

Aaron Clauset, Cosma Rohilla Shalizi, and Mark EJ Newman. 2009. Power-law distributions in empirical data. *SIAM review*, 51(4):661–703.

Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel Weld. 2020. SPECTER: Document-level representation learning using citation-informed transformers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2270–2282, Online. Association for Computational Linguistics.

Zhuyun Dai, Vincent Y Zhao, Ji Ma, Yi Luan, Jianmo Ni, Jing Lu, Anton Bakalov, Kelvin Guu, Keith B Hall, and Ming-Wei Chang. 2022. Promptagator: Few-shot dense retrieval from 8 examples. *arXiv preprint arXiv:2209.11755*.

Nouha Dziri, Ehsan Kamalloo, Sivan Milton, Osmar Zaiane, Mo Yu, Edoardo M. Ponti, and Siva Reddy. 2022. FaithDial: A faithful benchmark for information-seeking dialogue. *Transactions of the Association for Computational Linguistics*, 10:1473–1490.

Kenneth Enevoldsen, Márton Kardos, Niklas Muennighoff, and Kristoffer Nielbo. 2024. The scandinavian embedding benchmarks: Comprehensive assessment of multilingual and monolingual text embedding. In *Advances in Neural Information Processing Systems*.

Wenqi Fan, Yujuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. 2024. A survey on rag meeting llms: Towards retrieval-augmented large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '24, page 6491–6501, New York, NY, USA. Association for Computing Machinery.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 8342–8360, Online. Association for Computational Linguistics.

Helia Hashemi, Yong Zhuang, Sachith Sri Ram Kothur, Srivas Prasad, Edgar Meij, and W. Bruce Croft. 2023. Dense retrieval adaptation using target domain description. In *Proceedings of the 2023 ACM SIGIR International Conference on Theory of Information Retrieval*, ICTIR '23, page 95–104, New York, NY, USA. Association for Computing Machinery.

Dan Hendrycks, Steven Basart, Saurav Kadavath, Mantas Mazeika, Akul Arora, Ethan Guo, Collin Burns, Samir Puranik, Horace He, Dawn Song, et al. 2021. Measuring coding challenge competence with apps. *arXiv preprint arXiv:2105.09938*.

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*, pages 113–122.

Christoph Hoppe, David Pelkmann, Nico Migenda, Daniel Hötte, and Wolfram Schenck. 2021. Towards intelligent legal advisors for document retrieval and question-answering in german legal documents. In *2021 IEEE Fourth International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)*, pages 29–32.

Chao-Wei Huang, Chen-Yu Hsu, Tsu-Yuan Hsu, Chen-An Li, and Yun-Nung Chen. 2023. CONVERSER: Few-shot conversational dense retrieval with synthetic data generation. In *Proceedings of the 24th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 381–387, Prague, Czechia. Association for Computational Linguistics.

Junjie Huang, Duyu Tang, Linjun Shou, Ming Gong, Ke Xu, Daxin Jiang, Ming Zhou, and Nan Duan. 2021. CoSQA: 20,000+ web queries for code search and question answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5690–5700, Online. Association for Computational Linguistics.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Ali Shiraee Kasmaee, Mohammad Khodadad, Mohammad Arshi Saloot, Nicholas Sherck, Stephen Dokas, Hamidreza Mahyar, and Soheila Samiee. 2024. Chemteb: Chemical text embedding benchmark, an overview of embedding models performance & efficiency on a specific domain. *arXiv preprint arXiv:2412.00532*.

Tomáš Kočiský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2018. The NarrativeQA reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328.

Patrick Lewis, Barlas Oguz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. 2020a. MLQA: Evaluating cross-lingual extractive question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7315–7330, Online. Association for Computational Linguistics.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020b. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Minghan Li and Eric Gaussier. 2024. Domain adaptation for dense retrieval and conversational dense retrieval through self-supervision by meticulous pseudo-relevance labeling. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 5247–5259, Torino, Italia. ELRA and ICCL.

Xiangyang Li, Kuicai Dong, Yi Quan Lee, Wei Xia, Hao Zhang, Xinyi Dai, Yasheng Wang, and Ruiming Tang. 2025. CoIR: A comprehensive benchmark for code information retrieval models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 22074–22091, Vienna, Austria. Association for Computational Linguistics.

Zichao Li, Prakhar Sharma, Xing Han Lu, Jackie Cheung, and Siva Reddy. 2022. Using interactive feedback to improve the accuracy and explainability of question answering systems post-deployment. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 926–937, Dublin, Ireland. Association for Computational Linguistics.

Ye Liu, Rui Meng, Shafiq Joty, Silvio Savarese, Caiming Xiong, Yingbo Zhou, and Semih Yavuz. 2025. CodeXEmbed: A generalist embedding model family for multilingual and multi-task code retrieval. In *Second Conference on Language Modeling*.

Simon Lupart, Thibault Formal, and Stéphane Clinchant. 2023. Ms-shift: An analysis of ms marco distribution shifts on neural retrieval. In *Advances in Information Retrieval: 45th European Conference on Information*

Retrieval, ECIR 2023, Dublin, Ireland, April 2–6, 2023, Proceedings, Part I, page 636–652, Berlin, Heidelberg. Springer-Verlag.

Xueguang Ma, Minghan Li, Kai Sun, Ji Xin, and Jimmy Lin. 2021. Simple and effective unsupervised redundancy elimination to compress dense vectors for passage retrieval. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2854–2859, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Rui Meng, Ye Liu, Semih Yavuz, Divyansh Agarwal, Lifu Tu, Ning Yu, Jianguo Zhang, Meghana Bhat, and Yingbo Zhou. 2022. Augtriever: Unsupervised dense retrieval by scalable data augmentation. *arXiv preprint arXiv:2212.08841*.

Roshanak Mirzaee, Hossein Rajaby Faghihi, Qiang Ning, and Parisa Kordjamshidi. 2021. SPARTQA: A textual question answering benchmark for spatial reasoning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4582–4598, Online. Association for Computational Linguistics.

John X Morris and Alexander M Rush. 2024. Contextual document embeddings. *arXiv preprint arXiv:2410.02525*.

Niklas Muennighoff, Nouamane Tazi, Loic Magne, and Nils Reimers. 2023. MTEB: Massive text embedding benchmark. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2014–2037, Dubrovnik, Croatia. Association for Computational Linguistics.

Jianmo Ni, Gustavo Hernandez Abrego, Noah Constant, Ji Ma, Keith Hall, Daniel Cer, and Yinfei Yang. 2022. Sentence-t5: Scalable sentence encoders from pre-trained text-to-text models. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1864–1874, Dublin, Ireland. Association for Computational Linguistics.

Vikas Raunak, Vivek Gupta, and Florian Metze. 2019. Effective dimensionality reduction for word embeddings. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 235–243, Florence, Italy. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Mehrzad Shahinmoghadam and Ali Motamedi. 2024. Benchmarking pre-trained text embedding models in aligning built asset information. *arXiv preprint arXiv:2411.12056*.

Shamane Siriwardhana, Rivindu Weerasekera, Elliott Wen, Tharindu Kaluarachchi, Rajib Rana, and Suranga Nanayakkara. 2023. Improving the domain adaptation of retrieval augmented generation (rag) models for open domain question answering. *Transactions of the Association for Computational Linguistics*, 11:1–17.

Hongjin Su, Weijia Shi, Jungo Kasai, Yizhong Wang, Yushi Hu, Mari Ostendorf, Wen-tau Yih, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. 2023. One embedder, any task: Instruction-finetuned text embeddings. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1102–1121, Toronto, Canada. Association for Computational Linguistics.

Nandan Thakur, Nils Reimers, and Jimmy Lin. 2023. Injecting domain adaptation with learning-to-hash for effective and efficient zero-shot dense retrieval. In *Proceedings of the SIGIR 2023 Workshop on Reaching Efficiency in Neural Information Retrieval (ReNeuIR'23)*, Taipei, Taiwan.

Ellen Voorhees, Tasmeer Alam, Steven Bedrick, Dina Demner-Fushman, William R. Hersh, Kyle Lo, Kirk Roberts, Ian Soboroff, and Lucy Lu Wang. 2021. Trec-covid: constructing a pandemic information retrieval test collection. *SIGIR Forum*, 54(1).

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)*, pages 241–251, Melbourne, Australia. Association for Computational Linguistics.

David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020. Fact or fiction: Verifying scientific claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7534–7550, Online. Association for Computational Linguistics.

Kexin Wang, Nils Reimers, and Iryna Gurevych. 2021. TSDAE: Using transformer-based sequential denoising auto-encoderfor unsupervised sentence embedding learning. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 671–688, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Kexin Wang, Nandan Thakur, Nils Reimers, and Iryna Gurevych. 2022. GPL: Generative pseudo labeling for unsupervised domain adaptation of dense retrieval. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2345–2360, Seattle, United States. Association for Computational Linguistics.

Orion Weller, Benjamin Van Durme, Dawn Lawrie, Ashwin Paranjape, Yuhao Zhang, and Jack Hessel. 2024. Promptriever: Instruction-trained retrievers can be prompted like language models. *arXiv preprint arXiv:2409.11136*.

Marco Wrzalik and Dirk Krechel. 2021. GerDaLIR: A German dataset for legal information retrieval. In *Proceedings of the Natural Legal Language Processing Workshop 2021*, pages 123–128, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ikuya Yamada, Akari Asai, and Hannaneh Hajishirzi. 2021. Efficient passage retrieval with hashing for open-domain question answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 979–986, Online. Association for Computational Linguistics.

Weixiang Yan, Yuchen Tian, Yunzhe Li, Qian Chen, and Wen Wang. 2023. CodeTransOcean: A comprehensive multilingual benchmark for code translation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5067–5089, Singapore. Association for Computational Linguistics.

Steve Yang, Jason Rosenfeld, and Jacques Makutonin. 2018. Financial aspect-based sentiment analysis using deep representations. *arXiv preprint arXiv:1808.07931*.

Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. 2021. Jointly optimizing query encoder and product quantization to improve retrieval performance. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, CIKM '21, page 2487–2496, New York, NY, USA. Association for Computing Machinery.

Xin Zhang, Yanzhao Zhang, Dingkun Long, Wen Xie, Ziqi Dai, Jialong Tang, Huan Lin, Baosong Yang, Pengjun Xie, Fei Huang, Meishan Zhang, Wenjie Li, and Min Zhang. 2024. mGTE: Generalized long-context text representation and reranking models for multilingual text retrieval. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 1393–1412, Miami, Florida, US. Association for Computational Linguistics.

A General Trend for Eigenvalue distribution

The curves in Fig.3 are approximately linear on log–log axes, consistent with a power-law-like decay. To move beyond visual evidence, we quantify this trend across all evaluated datasets with the Sentence-T5 embeddings through the Kolmogorov-Smirnov (KS) test (Clauset et al., 2009).

Procedure: For each dataset, we form the sample covariance of mean-centered embeddings, $C = \frac{1}{n-1} X_c^\top X_c$, and obtain its eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots$. We then fit the tail of the spectrum to a rank–size power law

$$\lambda_k \approx C\, k^{-\beta}, \qquad k \geq k_{\min},$$

choosing $k_{\min}$ automatically to minimize the Kolmogorov-Smirnov (KS) distance between the empirical tail and its fitted curve, with a minimum tail length of 10 (i.e., the default lower bound (Clauset et al., 2009)). The decay exponent β is estimated by ordinary least squares (OLS) on log–log scales,

$$\log \lambda_k = a - \beta \log k + \varepsilon_k, \quad k \geq k_{\min},$$

and we report (i) the estimated β with an OLS-based 95% confidence interval, (ii) the log–log R^2, and (iii) a KS goodness-of-fit p-value obtained by parametric bootstrap under the fitted power law.

Summary statistics: Across all evaluated spectra, $\bar{\beta} = 82.716 \pm 7.909$, $R^2 = 0.971 \pm 0.012$, and every dataset passes the KS check with $p \geq 0.10$ (100%). These results indicate a consistently *steep* power-law–like decay of eigenvalues across datasets, supporting the practical use of aggressive low-rank compression: a relatively small number of leading components accounts for the bulk of the variance, while later components contribute negligibly.

Remarks: This analysis is descriptive rather than generative: eigenvalues are ordered and not independent, and the precise value of $\hat{\beta}$ can vary with the tail threshold and normalization choice. Nevertheless, the agreement between visual log–log linearity, and the uniformly good KS diagnostics makes the power-law interpretation a robust summary of the spectra observed; the two datasets in Fig. 3 serve as representative exemplars.

B Generalizability of Domain Semantics

To verify that our method captures generalizable domain semantics rather than merely overfitting to a specific set of test queries, we conducted a 3-fold cross-validation experiment.

For each dataset, we randomly split the queries into 3 folds. In each run, we fit the PCA only on the observed folds (simulating historical data) and evaluated on the held-out fold (unseen future queries). Table 4 presents the percentage improvement in NDCG@10 for Query Compression under this strict cross-validation setting.

The results are highly consistent with the results reported in the main paper. For example, NarrativeQA with GTE retains a ∼41% improvement even on unseen queries. This confirms that the principal components learned from a sample of queries successfully encode the underlying *domain structure* (user intent variance) rather than just memorizing the test set.

Dataset	MiniLM	GTE	Instr.	Dis-Ro	MP-All	MP-QA	Sent-T5	BGE	SFR
Apps	-2.6	0.5	-13.8	-13.3	1.3	-11.4	19.9	-14.9	-0.2
SciDocs	-1.3	19.3	2.8	-0.3	-1.1	2.9	3.3	4.3	2.0
MedQA	2.0	1.8	1.6	2.5	1.7	2.2	6.9	0.8	0.5
ArguAna	1.4	-1.7	2.5	0.3	-0.1	-0.2	10.9	2.4	-1.3
StackOverflow	0.1	-0.1	-1.1	0.4	-0.1	1.3	2.9	0.6	-0.9
TV2Nord	37.5	4.2	2.7	16.1	15.9	7.4	1.5	0.1	5.4
MLQA	0.4	0.6	2.2	0.7	1.0	0.6	0.8	0.2	1.8
NarrativeQA	10.9	41.8	5.6	16.4	13.5	5.8	20.9	2.2	3.0
SpartQA	537.4	124.2	304.7	456.9	1813.6	434.7	110.7	163.6	841.7

Table 4: NDCG@10 improvement (%) for Query Compression using 3-fold cross-validation. Positive gains are highlighted green, negative values are red. This setting evaluates the method on unseen queries to ensure generalization.

C Recall and Precision

Tables 5, 6 and Tables 7, 8 show the results for Recall@10 and Precision@10 similar to Table 1. Their performance gain patterns are almost identical with Table 1.

D Random Compression Baseline

To verify that the improvements in Table 1 are not by chance, we also experiment on 90% Random Compression (Table 9), where 10% of the original embedding dimensions are randomly chosen and removed. It confirms that Random Compression is unlikely to improve retrieval performance, and the gains from Query Compression are not by chance.

Datasets	Query Compression								
	MiniLM	GTE	Instr.	Dis-Ro	MP-QA	MP-All	Sent-T5	BGE	SFR
Code	1.0	0.8	-0.7	5.0	0.97	0.85	30.4	2.5	-0.1
Apps	-1.6	0.2	-12.3	-13.6	-11.4	2.75	13.7	-13.9	-0.3
SciDocs	-0.3	19.2	4.4	-0.4	2.7	-1.2	4.5	3.0	0.8
MedQA	1.7	1.2	1.0	1.7	2.0	1.4	5.7	1.6	0.5
ArguAna	0.4	-1.1	1.3	0.6	-0.2	-0.2	9.6	2.9	-0.6
StackOverflow	0.1	-0.3	-1.0	0.2	1.1	-0.1	3.2	0.1	-1.0
TV2Nord	33.4	3.3	2.5	15.9	7.4	15.1	1.6	0.1	5.6
GerDa	11.2	5.4	18.0	2.1	7.1	8.0	3.4	3.0	8.4
ARC	2.5	5.0	14.3	6.5	1.8	2.9	11.6	8.2	6.5
FeedbackQA	-0.9	0.7	-3.3	0.9	-2.7	-1.3	-1.5	-0.3	-0.2
FaithDial	0.9	4.8	5.2	-0.1	2.7	1.7	5.1	8.9	3.3
MLQA	0.5	0.8	2.1	0.7	0.6	1.0	-0.2	0.2	1.5
NarrativeQA	10.4	39.8	3.2	18.9	4.9	15.2	20.1	2.2	3.7
SpartQA	362.2	73.9	232.4	313.8	265.2	1207.1	106.8	116.5	696.3
Summary	11/14	12/14	10/14	11/14	11/14	10/14	12/14	12/14	9/14

Table 5: Recall@10 improvement (%) for Query Compression. Positive gains are highlighted green, negative values are red. This method enhances performance in 98 out of 126 data–model pairs (77.8%).

Datasets	Query+Document Compression								
	MiniLM	GTE	Instr.	Dis-Ro	MP-QA	MP-All	Sent-T5	BGE	SFR
Code	-0.7	-0.1	-1.2	0.5	-1.3	-1.0	16.1	-1.0	-0.3
Apps	-17.5	-1.3	-22.8	-48.1	-23.9	-11.0	4.5	-35.0	-1.3
SciDocs	-0.0	14.3	3.6	1.0	2.3	0.0	-4.6	-7.8	0.4
MedQA	1.6	1.0	1.3	1.6	1.6	0.9	4.6	1.4	0.6
ArguAna	0.1	-1.3	0.8	-0.3	-1.3	-1.5	6.3	1.6	-0.9
StackOverflow	0.0	-0.3	-1.1	-0.7	0.3	-0.1	3.2	-0.3	-1.0
TV2Nord	20.7	2.3	1.8	9.3	3.4	3.4	0.0	0.0	2.8
GerDa	1.1	-9.3	3.9	-8.4	-1.2	-2.1	-9.4	-11.9	-11.8
ARC	5.0	5.8	12.8	5.1	2.3	2.5	8.5	7.6	3.3
FeedbackQA	-1.9	0.8	-5.1	-6.7	-4.6	-3.1	-3.5	-0.7	-0.7
FaithDial	1.6	4.7	3.6	-0.1	1.6	1.8	4.1	8.2	3.4
MLQA	0.4	0.8	2.1	0.3	0.4	0.9	-1.3	0.4	1.2
NarrativeQA	7.0	35.9	0.5	14.5	-0.1	10.8	17.0	2.0	2.9
SpartQA	371.1	86.2	492.9	281.6	340.0	1035.7	214.9	118.1	713.8
Summary	9/14	9/14	10/14	8/14	8/14	7/14	9/14	7/14	8/14

Table 6: Recall@10 improvement (%) for Query+Document Compression. Positive gains are highlighted blue, negative values are red. This method enhances performance in 75 out of 126 data–model pairs (59.5%).

Datasets	Query Compression								
	MiniLM	GTE	Instr.	Dis-Ro	MP-QA	MP-All	Sent-T5	BGE	SFR
Code	0.9	0.8	-0.7	5.0	0.9	0.8	30.4	2.4	-0.1
Apps	-2.0	0.2	-12.3	-14.0	-11.2	2.4	14.1	-13.7	-0.3
SciDocs	-0.2	18.9	4.3	-0.4	2.8	-1.1	4.3	2.8	0.8
MedQA	1.6	1.2	1.0	1.7	1.9	1.4	5.7	1.5	0.5
ArguAna	0.5	-1.0	1.2	0.6	-0.1	-0.1	9.4	2.9	-0.6
StackOverflow	0.1	-0.3	-0.9	0.2	0.9	-0.1	3.1	0.1	-0.9
TV2Nord	33.3	3.3	2.3	15.7	7.5	14.9	1.6	0.0	5.5
GerDa	12.5	5.6	17.8	2.7	6.9	7.5	3.2	2.9	7.7
ARC	2.3	5.0	14.4	6.5	2.1	2.9	11.4	8.2	6.7
FeedbackQA	-0.9	0.7	-3.3	0.8	-2.6	-1.2	-1.5	-0.2	-0.2
FaithDial	0.8	4.7	5.3	-0.2	2.6	1.5	5.3	8.6	3.2
MLQA	0.5	0.7	2.0	0.6	0.5	1.0	-0.2	0.2	1.5
NarrativeQA	10.4	39.8	3.4	18.7	5.0	15.1	20.2	2.2	3.5
SpartQA	428.3	97.9	329.1	390.1	300.0	1433.3	84.0	148.0	784.6
Summary	11/14	12/14	10/14	11/14	11/14	10/14	12/14	11/14	9/14

Table 7: Precision@10 improvement (%) for Query Compression. Positive gains are highlighted green, negative values are red. This method enhances performance in 97 out of 126 data–model pairs (77.0%).

Datasets	Query+Document Compression								
	MiniLM	GTE	Instr.	Dis-Ro	MP-QA	MP-All	Sent-T5	BGE	SFR
Code	-0.7	0.0	-1.1	0.5	-1.2	-1.0	16.1	-1.0	-0.3
Apps	-17.1	-1.2	-22.4	-48.4	-24.1	-11.2	5.0	-35.0	-1.2
SciDocs	0.0	14.1	3.6	0.9	2.4	0.0	-4.6	-7.7	0.3
MedQA	1.6	1.0	1.3	1.6	1.5	0.8	4.5	1.3	0.5
ArguAna	0.1	-1.2	0.7	-0.2	-1.2	-1.4	6.2	1.7	-0.9
StackOverflow	0.0	-0.3	-1.0	-0.6	0.2	-0.1	3.1	-0.2	-1.0
TV2Nord	20.7	2.3	1.7	9.2	3.5	3.2	0.0	0.0	2.7
GerDa	2.5	-9.3	3.6	-6.9	-2.3	-1.5	-9.0	-11.7	-12.2
ARC	4.7	5.9	12.6	5.4	2.6	2.4	8.5	7.5	3.3
FeedbackQA	-1.9	0.8	-5.1	-6.7	-4.6	-3.0	-3.5	-0.7	-0.7
FaithDial	1.7	4.5	3.5	-0.2	1.5	1.7	4.2	7.9	3.4
MLQA	0.5	0.7	2.0	0.2	0.2	0.9	-1.3	0.3	1.2
NarrativeQA	7.2	35.8	0.5	14.1	0.0	10.6	17.0	2.0	2.9
SpartQA	456.6	115.0	697.9	373.7	373.5	1255.5	195.6	151.5	810.2
Summary	9/14	9/14	10/14	8/14	8/14	7/14	9/14	7/14	8/14

Table 8: Precision@10 improvement (%) for Query+Document Compression. Positive gains are highlighted blue, negative values are red. This method enhances performance in 75 out of 126 data–model pairs (59.5%).

E Low-Query Regime Analysis

The datasets in Table 10 and 11 with limited queries (typically much lower than the original 768-dimensional embeddings, as suggested by the table caption referring to scenarios where query count is less than the target PCA dimension) reveal PCA's adaptive robustness under Query Compression. The legal QA dataset *AILASt* shows strong positive gains across all nine listed models (ranging from +19.1% for `GTE` to +111.2% for `Dis-Ro`), indicating PCA effectively isolates structured variance even under severe rank constraints.

For datasets with extremely few queries, the challenge for PCA becomes more pronounced. For instance, the chemistry dataset *ChemNQ*, with only 27 queries, saw most models exhibit performance drops under Query Compression. This suggests that such a sparse dataset may not provide sufficient information for PCA to effectively identify and emphasize task-relevant variance across most models, with `Instr.` being an exception showing a marginal improvement (+3.7%).

Similarly, *COVID*, with only 50 queries, presented a challenging scenario. However, it's noteworthy that despite this limitation, a majority of models (five out of nine, including `GTE` +14.2%, `MiniLM` +8.9%, and `Instr.` +7.2%) still achieved performance gains with Query Compression. This highlights PCA's potential utility even when query data is scarce. Nevertheless, specific models like `Sent-T5` (-22.8%) still showed significant performance degradation, indicating that for some architectures or pre-training regimes, compression with very limited data can be detrimental if crucial high-

Datasets	90% Random Compression								
	MiniLM	GTE	Instr.	Dis-Ro	MP-All	MP-QA	Sent-T5	BGE	SFR
Code	-0.23	-0.18	-0.03	-0.12	-0.13	-0.02	-0.94	0.07	-0.18
Apps	-2.58	-0.23	-1.87	0.81	-0.83	-0.78	0.76	-2.17	-0.67
SciDocs	-1.2	-1.95	0.22	-0.09	-0.76	-0.75	-0.31	0.37	1.74
MedQA	0.65	-0.31	0.18	-0.22	0.68	0.02	-0.28	-0.65	-0.46
ArguAna	-0.48	-0.21	-1.04	-0.65	-0.8	-0.45	-1.29	-0.43	-0.99
StackOverflow	-0.48	-0.15	-0.09	-0.39	-0.13	-0.32	-0.12	-0.15	-0.73
TV2Nord	-3.78	-0.91	-0.52	-0.18	-0.69	-0.38	-0.38	-0.32	-1.21
GerDa	0	-1.5	-0.27	-0.23	-1.06	-0.39	-0.14	-0.5	-6.22
ARC	-0.53	-2.18	-0.72	-0.96	0.85	0.09	-0.12	-2.22	2.5
FeedbackQA	0.03	-0.49	-0.11	0.12	-0.37	-0.03	-0.73	-0.13	-0.65
FaithDial	-0.25	-0.35	-0.43	-0.12	0.41	-0.08	0.34	0.25	0.88
MLQA	-0.62	-0.33	-0.54	-0.44	-0.32	-0.26	-0.58	-0.54	1.51
NarrativeQA	-2.42	-0.96	-0.54	0.19	1.04	2.68	-2.43	-0.64	-0.69
SpartQA	3.03	-0.64	14.56	-16.76	-18.18	-3.94	8.17	8.54	-27.12
Summary	3/14	0/14	3/14	3/14	4/14	3/14	3/14	4/14	4/14

Table 9: NDCG@10 performance (%) using 90% Random Compression. Positive values are highlighted green and negative values are red. The "Summary" row indicates the number of datasets with improvements per model.

Datasets	Query Compression								
	MiniLM	GTE	Instr.	Dis-Ro	MP-QA	MP-All	Sent-T5	BGE	SFR
CosQA	1.3	-1.6	0.7	10.4	4.5	6.3	7.0	7.7	7.0
SciFact	-0.4	-0.1	1.4	-0.8	-1.2	0.2	2.1	1.7	-2.1
NFCorpus	-2.7	0.3	-0.1	-2.8	-4.7	-1.9	0.5	-1.1	-0.2
FiQA	0.0	-0.2	2.5	-1.0	-0.6	-0.1	0.3	-1.0	-2.2
BuiltBench	0.1	1.6	3.6	1.5	2.3	2.7	2.9	4.3	6.3
Legal	4.7	24.4	22.3	1.8	27.9	9.1	8.9	14.6	20.0
AILACDocs	19.7	29.8	0.4	20.3	28.4	14.7	11.5	8.5	31.5
CodeTrans	-6.4	-10.9	-3.2	-1.5	0.7	-0.5	-8.2	-0.8	-5.1
AILASt	40.7	19.1	53.6	111.2	58.4	70.9	51.9	32.0	63.5
COVID	8.9	14.2	7.2	-7.5	0.8	2.8	-22.8	-6.9	-11.9
ChemNQ	-10.8	-18.7	3.7	-14.8	-19.5	-19.6	-14.5	-24.7	-11.2
Summary	6/11	6/11	9/11	5/11	7/11	7/11	7/11	6/11	5/11

Table 10: NDCG@10 improvement (%) comparison for Query Compression for datasets where the number of queries is lower than most model's target PCA dimension. Positive gains are highlighted green, negative values are red. Summary rows show the number of datasets with improvements per model.

dimensional interactions are disrupted.

F Detailed NDCG@10 Results for Query Compression

Table 1 reports only the *relative* NDCG@10 changes (%) from Query Compression. For completeness, Table 12 and 13 provides the corresponding *absolute* NDCG@10 values, with the relative change shown in parentheses.

G SpartQA

SpartQA is not a typical retrieval dataset, where in MTEB's setting, the multiple-choice answers for each query are all dumped together, forming a corpus of 1.5k instances. Without PCA, some of the models initially have nearly zero NDCG@10. This is because without knowing that SpartQA focuses on spatial reasoning and that questions needs to be distinguished by the objects and colors occurred, the semantics of texts in SpartQA are highly similar and therefore less discriminative (i.e., all about spatial relations, colors, and shapes).

Datasets	Query+Document Compression								
	MiniLM	GTE	Instr.	Dis-Ro	MP-QA	MP-All	Sent-T5	BGE	SFR
CosQA	2.5	-1.5	5.0	4.1	8.8	-0.4	-1.2	-2.9	0.7
SciFact	-1.3	0.4	0.5	-2.9	-2.3	-0.9	-1.1	-0.8	-1.2
NFCorpus	-7.8	0.0	-3.9	-9.9	-3.3	-8.5	-4.8	-4.4	-5.1
FiQA	-0.2	0.1	2.9	-0.7	-0.5	0.1	-2.2	-1.6	-1.2
BuiltBench	0.0	-0.3	0.6	0.4	-0.4	0.2	-1.2	-0.6	1.5
Legal	6.2	10.1	-5.5	-11.1	-4.6	4.3	-11.6	7.1	-3.5
AILACDocs	12.2	26.1	4.2	16.3	12.2	25.1	8.9	-1.3	22.7
CodeTrans	-4.8	0.4	-2.7	4.4	1.6	1.6	-2.2	1.5	-0.2
AILASt	11.9	-0.3	23.8	38.6	38.5	29.1	5.4	7.1	32.8
COVID	-2.1	16.9	7.0	-4.2	-2.6	-2.0	-27.2	-20.9	-2.5
ChemNQ	-0.0	10.3	2.3	-3.6	0.3	1.7	1.6	-5.0	1.2
Summary	4/11	7/11	8/11	5/11	5/11	7/11	3/11	3/11	5/11

Table 11: NDCG@10 improvement (%) comparison for Query+Document Compression for datasets where the number of queries is lower than most models' target PCA dimension. Positive gains are highlighted blue, negative values are red. Summary rows show the number of datasets with improvements per model.

H Failure Analysis

Query Level Effects We present an example where PCA compression reduces the performance. On *FeedbackQA + Sent-T5* encoder, the NDCG@10 drops by **2.4%**. Although the loss is small, a closer look reveals a nuanced redistribution of performance rather than a uniform degradation. There are 119 queries improved (the ground-truth document was ranked higher after PCA), while 179 queries regressed (the ground-truth document was ranked lower after PCA). Hence, PCA reshapes the embedding space in a way that facilitates retrieval on a non-trivial subset of queries even while slightly hurting the mean.

Similarity-Score Distributions We compared cosine-similarity distributions between queries and their *relevant* (ground truth) versus *non-relevant* (hard-negative) documents before and after PCA (See Table 14)

PCA increases the mean gap between relevant and non-relevant scores (better class separation), but it also inflates the standard deviation for both classes. The heavier tails create a larger overlap region near the decision boundary, introducing ambiguity for borderline documents. These additional borderline cases explain the modest overall decline in NDCG: PCA resolves some rank-ordering errors but simultaneously introduces new ones. The

Dataset	MiniLM	GTE	Instr.	Dis-Ro	MP-All	MP-QA	Sent-T5	BGE	SFR
Apps	6.3 (-4.1)	58.3 (+1.1)	7.8 (-14.5)	3.1 (-16.2)	8.5 (+1.1)	6.8 (-11.1)	7.7 (+18.0)	12.4 (-16.1)	49.5 (-0.2)
SciDocs	21.6 (-0.2)	23.1 (+18.5)	19.1 (+3.0)	21.6 (-0.5)	23.5 (-1.1)	18.0 (+3.3)	16.6 (+3.8)	17.1 (+4.8)	25.9 (+2.0)
MedQA	61.6 (+2.1)	72.9 (+2.0)	68.8 (+2.1)	61.9 (+2.7)	68.0 (+2.2)	67.7 (+1.9)	65.0 (+6.9)	68.4 (+0.4)	75.1 (+0.5)
ArguAna	50.8 (+1.2)	72.8 (-2.2)	57.2 (+3.0)	48.2 (+0.6)	46.6 (+0.3)	50.9 (-0.3)	43.8 (+11.1)	55.2 (+2.2)	69.9 (-1.3)
StackOverflow	84.0 (+0.1)	91.1 (-0.1)	84.0 (-1.1)	72.1 (+0.3)	90.2 (-0.1)	82.6 (+1.2)	83.8 (+2.9)	80.9 (+0.4)	89.1 (-0.9)
TV2Nord	47.8 (+40.2)	74.5 (+4.4)	75.8 (+2.8)	45.8 (+17.1)	49.5 (+17.7)	56.9 (+7.5)	69.1 (+1.2)	94.6 (+0.3)	72.2 (+5.5)
GerDa	2.6 (+6.2)	15.2 (+3.5)	13.0 (+15.7)	4.2 (-1.4)	4.1 (+7.4)	5.5 (+7.0)	7.2 (+3.2)	32.6 (+1.8)	15.4 (+7.8)
ARC	9.7 (+2.5)	13.5 (+1.4)	14.1 (+12.9)	10.7 (+2.7)	12.2 (+3.0)	11.1 (+2.0)	18.0 (+11.1)	9.7 (+7.3)	14.2 (+4.4)
FeedbackQA	57.2 (-1.9)	70.6 (+1.0)	62.7 (-3.7)	48.8 (+0.2)	58.1 (-2.2)	60.3 (-3.2)	62.8 (-2.4)	69.0 (-0.4)	71.0 (-0.2)
FaithDial	24.1 (+0.3)	24.2 (+5.1)	24.0 (+4.0)	24.5 (+0.8)	24.7 (+0.8)	24.9 (+1.8)	24.2 (+4.2)	21.7 (+7.9)	26.2 (+4.8)
Code	78.4 (+1.2)	90.2 (+2.3)	90.7 (-1.0)	82.4 (+6.0)	78.1 (+1.1)	81.9 (+1.3)	87.3 (+38.7)	92.5 (+4.0)	97.3 (-0.2)
MLQA	61.4 (+0.5)	70.3 (+0.6)	67.7 (+2.5)	61.9 (+0.8)	62.7 (+1.0)	64.6 (+0.4)	60.6 (-0.4)	70.9 (+0.2)	68.8 (+1.8)
NarrativeQA	20.2 (+11.0)	42.6 (+41.5)	27.8 (+7.1)	18.3 (+15.3)	21.8 (+13.1)	21.7 (+5.8)	24.8 (+20.5)	49.5 (+1.6)	40.2 (+2.5)
SpartQA	11.9 (+620.0)	19.5 (+149.5)	7.0 (+343.0)	10.9 (+508.4)	4.5 (+1954.5)	7.8 (+511.0)	4.6 (+119.7)	21.9 (+192.5)	11.2 (+849.1)

Table 12: Raw NDCG@10 under Query Compression, with the relative change from the uncompressed baseline shown in parentheses (%). This table provides the absolute values corresponding to the improvement-only summary in Table 1. Positive gains are highlighted green, negative values are red.

Datasets	Query+Document Compression								
	MiniLM	GTE	Instr.	Dis-Ro	MP-All	MP-QA	Sent-T5	BGE	SFR
Apps	5.2 (-21.1)	56.8 (-1.5)	6.8 (-24.8)	1.8 (-52.8)	7.4 (-12.4)	6.0 (-22.1)	6.9 (+5.0)	9.3 (-36.8)	49.0 (-1.1)
SciDocs	21.6 (-0.4)	22.0 (+13.3)	19.0 (+2.5)	21.8 (+0.5)	23.7 (-0.3)	17.8 (+2.4)	15.1 (-5.2)	15.0 (-7.9)	25.8 (+1.9)
MedQA	61.5 (+1.9)	72.5 (+1.4)	68.5 (+1.6)	61.6 (+2.2)	67.4 (+1.3)	67.0 (+0.9)	64.2 (+5.6)	67.8 (-0.4)	75.0 (+0.5)
ArguAna	50.4 (+0.4)	72.6 (-2.5)	56.4 (+1.6)	47.8 (-0.4)	46.0 (-1.1)	50.3 (-1.4)	42.2 (+7.1)	54.2 (+0.4)	69.4 (-2.0)
StackOverflow	84.1 (+0.1)	91.2 (+0.0)	84.0 (-1.1)	71.3 (-0.8)	90.3 (-0.1)	82.0 (+0.5)	83.8 (+2.9)	80.5 (-0.1)	89.0 (-1.1)
TV2Nord	42.6 (+24.9)	73.3 (+2.7)	75.6 (+2.6)	43.7 (+11.7)	43.8 (+4.2)	55.1 (+4.0)	67.9 (-0.6)	94.1 (-0.2)	70.2 (+2.7)
GerDa	2.4 (-1.7)	13.2 (-10.1)	11.4 (+2.0)	3.8 (-10.8)	3.7 (-1.6)	5.1 (-0.6)	6.4 (-7.9)	27.5 (-14.2)	12.4 (-13.4)
ARC	9.8 (+3.7)	13.5 (+1.7)	14.1 (+13.1)	10.6 (+2.3)	12.3 (+3.8)	11.1 (+1.9)	17.7 (+9.4)	9.8 (+8.2)	14.0 (+3.0)
FeedbackQA	56.7 (-2.7)	70.2 (+0.5)	61.2 (-6.0)	45.4 (-6.9)	57.3 (-3.5)	59.0 (-5.3)	61.4 (-4.6)	68.5 (-1.2)	70.3 (-1.2)
FaithDial	24.4 (+1.3)	24.2 (+4.8)	23.6 (+2.3)	24.3 (-0.2)	24.9 (+1.4)	24.7 (+0.8)	24.0 (+3.1)	21.7 (+7.9)	26.2 (+4.6)
Code	76.9 (-0.7)	89.0 (+1.0)	90.0 (-1.8)	78.3 (+0.7)	76.5 (-1.0)	79.8 (-1.3)	74.8 (+18.8)	87.8 (-1.2)	96.9 (-0.6)
MLQA	61.2 (+0.2)	70.3 (+0.6)	67.7 (+2.5)	61.6 (+0.4)	62.6 (+0.8)	64.5 (+0.3)	59.8 (-1.6)	71.0 (+0.3)	68.7 (+1.7)
NarrativeQA	19.5 (+7.1)	41.3 (+37.0)	27.2 (+4.7)	17.7 (+11.6)	20.9 (+8.5)	20.8 (+1.5)	24.1 (+17.4)	49.2 (+1.1)	39.8 (+1.6)
SpartQA	12.0 (+626.7)	21.4 (+173.4)	12.9 (+717.7)	10.1 (+463.1)	3.5 (+1509.1)	9.3 (+632.3)	7.2 (+248.1)	23.1 (+208.9)	11.0 (+829.7)

Table 13: Raw NDCG@10 under Query+Document Compression, with the relative change from the uncompressed baseline shown in parentheses (%). This table provides the absolute values corresponding to the improvement-only summary in Table 1. Positive gains are highlighted blue, negative values are red.

	Original	PCA-compressed
Relevant	μ=0.8361, σ=0.0460	μ=0.4700, σ=0.1402
Non-relevant	μ=0.6923, σ=0.0546	μ=0.0038, σ=0.1236

Table 14: Cosine Similarity distribution for relevant and non-relevant documents before and after PCA.

mixed outcome underscores that PCA's impact is query-specific: it can sharpen the signal for certain queries while blurring it for others.

I Dataset Details

The 14 datasets from MTEB with sufficient queries are: `COIRCodeSearchNetRetrieval` (python subset) (Li et al., 2025), `AppsRetrieval` (Hendrycks et al., 2021), `ArguAna` (Wachsmuth et al., 2018), `MedicalQARetrieval` (Ben Abacha and Demner-Fushman, 2019), `SciDocs`(Cohan et al., 2020), `StackOverflowQA` (Li et al., 2025), `TV2Nord` (Enevoldsen et al., 2024), `ARC` (Clark et al., 2018), `GerDa` (Wrzalik and Krechel, 2021), `FeedbackQA` (Li et al., 2022), `FaithDial` (Dziri et al., 2022), `NarrativeQA` (Kočiský et al., 2018), `MLQA` (eng-eng subset) (Lewis et al., 2020a), and SpartQA (Mirzaee et al., 2021).

The 11 datasets with not enough queries are: `SciFact` (Wadden et al., 2020), `ChemNQRetrieval` (Kasmaee et al., 2024), `FiQA2018` (Yang et al., 2018), `NFCorpus` (Boteva et al., 2016), `TRECCOVID` (Voorhees et al., 2021), `Legal` (Hoppe et al., 2021), `CodeTransDL` (Yan et al., 2023), `CosQA` (Huang et al., 2021),

Dataset	#Queries	#Docs	Language	Content Category
AILACasedocs	50	186	English	Legal Case Documents
CodeTransOceanDL	180	816	Code	Code Translation
CosQA	500	20604	English	Code QA
StackOverflowQA	1994	19931	English	Programming Q&A
TV2NordRetrieval	2048	2048	Danish	News Articles
GerDaLIRSmall	12234	9969	German	Legal Documents
LegalQuAD	200	200	German	Legal Q&A
AILAStatutes	50	82	English	Legal Statutes
ARCChallenge	1172	9350	English	Science QA
BuiltBenchRetrieval	334	2761	English	Engineering Retrieval
FeedbackQARetrieval	1992	2364	English	Public Service QA
MedicalQARetrieval	2048	2048	English	Medical QA
ChemNQRetrieval	27	22933	English	Chemistry QA
AppsRetrieval	3765	8765	English	Code Snippet Retrieval
FiQA2018	648	57638	English	Finance QA
NFCorpus	323	3633	English	Biomedical Abstracts
SCIDOCS	1000	25657	English	Scientific Papers
SciFact	300	5183	English	Scientific Fact Verification
COIRCodeSearchNetRetrieval	14918	280310	Code	Code Search Retrieval
ArguAna	1406	8674	English	Argumentative Texts
TRECCOVID	50	171332	English	COVID-19 QA
MLQARetrieval	11582	9916	English	General QA
NarrativeQARetrieval	10557	355	English	Narrative QA
SpartQA	3594	1592	English	Spatial Reasoning QA
FaithDial	2042	3539	English	Dialogue

Table 15: Overview of retrieval datasets: number of queries and documents, primary language, and content category.

`BuiltBench` (Shahinmoghadam and Motamedi, 2024), `AILAStat` and `AILADocs` (Bhattacharya et al., 2019).

More details on each dataset has been given in Table 15.

J Model Details

Abbrev.	Full model name
Sentence-Transformers family (Reimers and Gurevych, 2019)	
`Dis-Ro`	`all-distilroberta-v1`
`MP-QA`	`multi-qa-mpnet-base-dot-v1`
`MiniLM`	`all-MiniLM-L6-v2`
`MP-All`	`all-mpnet-base-v2`
Other embedding models	
`Sent-T5`	`sentence-t5-xl` (Ni et al., 2022)
`BGE`	`BGE_M3` (Chen et al., 2024)
`GTE`	`GTE` (Zhang et al., 2024)
`SFR`	`SFRCode` (Liu et al., 2025)
`Instr.`	`Instructor` (Su et al., 2023)

Table 16: Pretrained dense retrieval models evaluated in this work.

Table 16 lists the pretrained dense retrieval models used in our evaluation, along with the abbreviations used throughout the paper.