Unveiling DIME: Reproducibility, Generalizability, and Formal Analysis of Dimension Importance Estimation for Dense Retrieval

Cesare Campagnano
Pinecone
New York, US
cesare@pinecone.io

Antonio Mallia
Pinecone
New York, US
antonio@pinecone.io

Fabrizio Silvestri
Sapienza University of Rome
Rome, Italy
fsilvestri@diag.uniroma1.it

Abstract

Dimension IMportance Estimation (DIME) is a recently proposed technique to enhance ranking effectiveness of dense retrieval models by pruning irrelevant embedding dimensions through Pseudo Relevance Feedback (PRF DIME) or exploiting dense representations of Large Language Model-generated answers (LLM DIME). Despite strong empirical performance, its theoretical foundations and generalizability remain open questions.

In this paper, we propose four key contributions. First, we provide a rigorous theoretical analysis of DIME, framing it as a denoising mechanism that mitigates embedding noise while preserving the salient information. Second, we conduct a comprehensive reproducibility study, confirming previously reported gains for both PRF DIME and LLM DIME. Third, we extend the evaluations of PRF DIME by applying it to a broader set of embedding models with distinct characteristics, such as matryoshka embeddings, cosine similarity-optimized models, and architectures that produce high-dimensional representations, while also testing it on diverse retrieval datasets. For LLM DIME, we expand the analysis across a range of LLMs, comparing high-parameter proprietary models with cheaper open-source alternatives. Finally, we refine DIME by introducing an attention-inspired PRF mechanism and propose to leverage dimension importance as a reranking technique.

https://github.com/pinecone-io/unveiling-dime

CCS Concepts

• **Information systems → Query representation**; **Retrieval models and ranking**; **Document representation**.

Keywords

Dense Retrieval, Dimension Importance Estimation, Denoising

ACM Reference Format:
Cesare Campagnano, Antonio Mallia, and Fabrizio Silvestri. 2025. Unveiling DIME: Reproducibility, Generalizability, and Formal Analysis of Dimension Importance Estimation for Dense Retrieval. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '25), July 13–18, 2025, Padua, Italy*. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3726302.3730318

ACM ISBN 979-8-4007-1592-1/2025/07
https://doi.org/10.1145/3726302.3730318

1 Introduction

Embedding-based neural retrieval models now lie at the heart of modern Information Retrieval (IR), mapping queries and documents into high-dimensional vector spaces for contextual matching [20, 29]. Unlike older sparse methods such as BM25 [30] or even current learned sparse models [5, 11, 19, 21–23, 45], dense embeddings deliver remarkable improvements by capturing higher-level semantic similarities. However, the move to high-dimensional embedding spaces introduces new challenges. Not all coordinates contribute equally to capturing relevance signals, and certain dimensions can even hurt retrieval accuracy by introducing noise.

In response, recent work has explored dimension pruning to mitigate these issues and to focus on the most salient coordinates for a particular query. One such approach is the *Dimension IMportance Estimation* (DIME) framework [10], which proposes selecting a *query-dependent* subspace in which only the most relevant dimensions are retained. Empirically, DIME has shown to significantly improve ranking effectiveness across multiple benchmarks.

Despite these promising results, some key research questions remain open. First, while DIME has demonstrated strong empirical performance, its mathematical underpinnings remain unclear. A rigorous *theoretical analysis* of DIME would clarify how dimension selection balances discarding noise and preserving information. By casting dimension pruning in a *denoising* framework, we could better understand the geometry of subspace selection and why removing certain coordinates boosts retrieval performance (**RQ1**). Second, *reproducibility* of DIME has yet to be fully established. As with other neural IR methods, small changes to hyperparameters or evaluation procedures may yield divergent findings [2] (**RQ2**). Third, DIME's *generalizability* to newly emerging embedding models has not been systematically investigated. Modern IR applications involve both extremely large-scale embeddings (potentially exacerbating dimensionality issues) and compact, efficiency-driven encoders designed for real-world deployment. Whether DIME remains effective across different datasets, domains, and embedding architectures is an open question (**RQ3**). Finally, the potential for refining DIME to further enhance its applicability and performance remains unexplored (**RQ4**).

In this paper, we address these open questions through a combination of theoretical insights, reproducibility analysis, generalization studies, and methodological refinements. To improve readability, we structure the remainder of this work to directly align with these research questions. Section 2 provides an overview of key concepts related to dense retrieval, pseudo-relevance feedback (PRF), large language models, and the DIME framework. Section 3 addresses **RQ1** by establishing a formal foundation for dimension pruning, framing it as a denoising mechanism to explain its effectiveness. Section 4 examines **RQ2**, presenting a comprehensive reproducibility study to assess the robustness of DIME's empirical improvements. Section 5 investigates **RQ3**, evaluating the generalizability of DIME across different retrieval scenarios, embedding models, and datasets. Finally, Section 6 explores **RQ4**, proposing two refinements: (i) a *weighted averaging* technique that improves retrieval effectiveness while making the model more robust to hyperparameter selection, and (ii) a *reranking* strategy that leverages DIME to refine ranking results, enhancing applicability. Section 7 summarizes our findings and discusses future directions.

2 Background and Preliminaries

This section provides the necessary background for understanding our work. We begin with an overview of dense vector retrieval, followed by a recap of pseudo-relevance feedback (PRF) approaches. We then summarize current developments in LLMs and, finally, introduce the Dimension IMportance Estimation (DIME) paradigm.

Dense Vector Retrieval. Over the past decade, the paradigm of *vector retrieval* has rapidly become a dominant approach in IR, propelled by advances in neural networks and LLMs. Traditional IR methods such as BM25 rely primarily on lexical overlap, treating each unique token as a dimension in a large sparse vector space. While these methods are robust and interpretable, they can be limited when query and document vocabulary differs (e.g., synonyms or paraphrases). By contrast, *dense embedding models* project queries and documents into a shared, lower-dimensional representation space [8]. Each textual input is mapped to a dense vector, typically of a few hundred dimensions (e.g., 768), by a neural encoder. Retrieval then proceeds by measuring similarity between the query and document embeddings. Common *retrieval metrics* for dense representations include Inner Product (IP) and Cosine Similarity.

Normalization. In many practical systems, *normalizing* all embeddings to unit length (i.e., $\|\mathbf{x}\|_2 = 1$) is performed so that an *inner product* between normalized vectors *directly* corresponds to their cosine similarity. Such normalization can simplify the search pipeline, since one only needs to store *unit* vectors and use a standard dot-product index. Moreover, it ensures that each embedding's magnitude does not unfairly dominate the distance measure, focusing instead on the *directional* alignment in the latent space.

Matryoshka Embeddings. Traditional embedding models produce fixed-size vector representations, often without consideration for task-specific requirements or scalability. *Matryoshka Representation Learning* [17] addresses this limitation by enabling embeddings to remain effective even when truncated to smaller dimensions. This is achieved by applying the same training objective to both the full embedding and its truncated portions.

Pseudo-Relevance Feedback. PRF is a classical technique in IR that refines a query representation using the top-ranked documents initially retrieved for that query. The assumption is that these top documents are *likely* to be relevant, even if they are not manually labeled. Rocchio [31] introduced a pioneering vector-space method for iteratively adjusting the query based on these feedback documents, while later work like RM3 used statistical term-level expansions. In dense retrieval, PRF is often interpreted as *shifting* the query embedding toward the centroid of its top-k retrieved embeddings, thereby injecting relevance signals to improve subsequent ranking.

Large Language Models (LLMs). With the advent of the Transformer architecture [35], language modeling shifted from bidirectional encoders such as BERT [8] to autoregressive causal decoders like GPT-2 and GPT-3 [6, 26, 27]. These *generative models* allow for flexible text generation at large scales and now include both closed-source variants and open-source models suitable for production.

Empirical work on scaling laws [16, 28] shows that increasing model size typically boosts performance. Proprietary models such as GPT-4 and GPT-4o [1] lead many benchmarks, but open-source LLMs like Llama [9, 33, 34] and Qwen [3, 43, 44] can closely compete, especially with specialized training or domain adaptation. Moreover, recent attempts to optimize inference-time compute via reinforcement learning have led to the OpenAI o1/o3 series [15] and DeepSeek R1 [12], each designed to improve step-by-step reasoning, resulting in an overall higher-quality output.

Despite strong performance, large models often incur high inference costs. GPT-4 may require tens of seconds to generate a complex answer, making it challenging in real-time retrieval scenarios. Hence, while we demonstrate how GPT-4 can produce synthetic documents for *Dimension Importance Estimation* (Section 2), we also investigate smaller or more cost-effective LLMs (Qwen 2.5, Llama 3.x, DeepSeek R1) that offer faster inference.

Dimension Importance Estimators (DIME). Although dense retrieval brings clear advantages, recent evidence shows that *not all* dimensions in a latent space are equally useful for every query [10]. A fixed-size vector (e.g., dimension 768) can encode many linguistic or conceptual features, but some coordinates may be irrelevant—or even detrimental—for a specific information need. This reality is further exacerbated when embeddings exhibit hierarchical (*Matryoshka*) structures or when the vectors vary widely in norm.

Key Concept. DIME assigns an importance score $u_q(i)$ to each coordinate $i \in \{1, \ldots, d\}$ of the query embedding $\mathbf{q} \in \mathbb{R}^d$. Sorting dimensions by $u_q(i)$ and retaining the top-k effectively *projects* the query onto a query-adaptive subspace. Empirical findings indicate that dimension filtering can substantially improve retrieval metrics while requiring neither re-training nor re-indexing [10].

Magnitude DIME. One of the simplest DIMEs uses only the absolute value of each query coordinate:

$$u_q^{\mathrm{mag}}(i) \;=\; \left|q_i\right|.$$

This method assumes larger components in the query embedding carry greater importance. Though computationally cheap, it overlooks crucial "small" coordinates that are semantically relevant [10].

PRF DIME. Drawing on the PRF concept (Section 2), suppose $d_1, \ldots, d_{k_f}$ are the top-k_f (pseudo-relevant) documents retrieved for q. Their centroid is

$$\mathbf{p} = \frac{1}{k_f} \sum_{j=1}^{k_f} \mathbf{d}_j.$$

The **PRF DIME** then scores each dimension by

$$u_q^{PRF@k_f}(i) \;=\; q_i \,\cdot\, p_i,$$

i.e., coordinates of $\mathbf{q}$ that align strongly with the centroid of pseudo-relevant embeddings are prioritized.

LLM DIME. Finally, **LLM DIME** utilizes a generative model to produce a synthetic passage $\mathbf{a} \in \mathbb{R}^d$:

(1) Prompt an LLM (e.g., GPT-4) with query q,
(2) Encode the generated text into $\mathbf{a}$,
(3) Score each dimension via $u_q^{LLM}(i) = q_i \cdot a_i$.

This approach can be powerful in scenarios where top-retrieved documents are suboptimal or where we can afford the cost of LLM calls. It can yield strong improvements in zero-shot or domain-mismatch tasks, though it may suffer from high inference latency if large LLMs are used at scale.

Recent work [7] has also shown the importance of negative feedback in the dimension importance estimation.

3 DIME from a Denoising Perspective

Dimension IMportance Estimation (DIME) works by selectively pruning embedding coordinates deemed irrelevant to a query, thereby focusing on dimensions most indicative of relevance. In this section, we argue that this process can be interpreted as a form of *denoising*, where dimension filtering acts to suppress random fluctuations in embedding vectors while preserving the core semantic "signal" that corresponds to the user's information need.

3.1 Signal vs. Noise in Top-Ranked Documents

The term *information need* is common in information science and describes a person's or a group's wish to find and get information that meets a conscious or unconscious requirement. Hjørland [40] explains that it is closely linked to relevance: if something is important for someone to complete a specific task, we can say that person needs that information.

Let $\{\mathbf{d}^{(i)}\}_{i=1}^k \subset \mathbb{R}^h$ be the set of top-k retrieved documents (in embedding space). We model each $\mathbf{d}^{(i)}$ as a sum of a *signal* component plus a *noise* term:

$$\mathbf{d}^{(i)} = \alpha^{(i)}\mathbf{s} + \boldsymbol{\epsilon}^{(i)},$$

where

- $\mathbf{s} \in \mathbb{R}^h$ is the *information need* (the signal) common to the top-k retrieved documents,
- $\alpha^{(i)} \geq 0$ reflects how strongly document i expresses that information need , and
- $\boldsymbol{\epsilon}^{(i)} \in \mathbb{R}^h$ is the noise component specific to document i.

Furthermore, following the usual convention, a subscript j to a vector represents the j-th component of that vector, thus component-wise we write

$$\mathbf{d}_j^{(i)} = \alpha^{(i)}\mathbf{s}_j + \boldsymbol{\epsilon}_j^{(i)},$$

We assume

$$\mathbb{E}[\boldsymbol{\epsilon}^{(i)}] = \mathbf{0}, \quad \mathrm{Var}\big(\epsilon_j^{(i)}\big) = \sigma^2,$$

and that $\epsilon_j^{(i)}$ is independent across indices i, j. The scalar $\alpha^{(i)}$ and the vector $\mathbf{s}$ are unobserved but serve to illustrate that the top-k embeddings share a common "core signal" plus random noise $\boldsymbol{\epsilon}^{(i)}$.

Signal Concentration. A standard way to aggregate the top-k embeddings is via a weighted centroid:

$$\mathbf{c} = \sum_{i=1}^k w^{(i)}\,\mathbf{d}^{(i)} \quad \text{where} \quad \sum_{i=1}^k w^{(i)} = 1.$$

Writing c_j as the j-th dimension of $\mathbf{c}$, we have

$$c_j = \sum_{i=1}^k w^{(i)}\big[\alpha^{(i)}s_j + \epsilon_j^{(i)}\big] = s_j \sum_{i=1}^k w^{(i)}\alpha^{(i)} + \underbrace{\sum_{i=1}^k w^{(i)}\,\epsilon_j^{(i)}}_{\eta_j}. \quad (1)$$

Thus, in the special case of uniform weighting, $w^{(i)} = \frac{1}{k}$,

$$\eta_j = \frac{1}{k}\sum_{i=1}^k \epsilon_j^{(i)}, \quad \mathbb{E}[\eta_j] = 0, \quad \mathrm{Var}(\eta_j) = \frac{\sigma^2}{k}.$$

Viewing Relevance as Dot Product. Let $\mathbf{q} \in \mathbb{R}^h$ be a query embedding and $r(\cdot)$ be our relevance scoring function that we assume to be computed by simple dot-product, we have

$$r\Big(q, d^{(i)}\Big) = q \cdot d^{(i)} = \sum_{j=1}^h q_j d_j^{(i)} = \sum_{j=1}^h \Big(q_j\alpha^{(i)}s_j + q_j\epsilon_j^{(i)}\Big)$$

The contribution of the j-th term of the query to $r(\cdot)$ is given by:

$$r_j\left(q_j, d_j^{(i)}\right) = q_j\alpha^{(i)}s_j + q_j\epsilon_j^{(i)}$$

Likewise, the dot product with the centroid becomes

$$r(q,c) = \sum_{j=1}^h q_j\,c_j = \sum_{j=1}^h q_j \sum_{i=1}^k w^{(i)}\,\alpha^{(i)}\,s_j + \sum_{j=1}^h q_j\,\eta_j$$

Optimal Hard Thresholding (Masking Dimensions). In DIME, a binary mask $\mathbf{m} = (m_1, m_2, \ldots, m_h) \in \{0,1\}^h$ is learned to selectively turn off dimensions deemed less relevant to $\mathbf{q}$. Concretely, $m_j = 0$ means we zero out dimension j in subsequent dot-product similarity computations; $m_j = 1$ means the dimension is retained.

A natural (though idealized) objective is to keep as much of the signal as possible while suppressing noise. For instance, one might minimize

$$\min_{\mathbf{m}\in\{0,1\}^h} \left[\left|\sum_{j=1}^h m_j\,q_j\,\widehat{s}_j - \sum_{j=1}^h q_j\,s_j\right| + \lambda \sum_{j=1}^h m_j\,q_j\,\eta_j\right], \quad (2)$$

where

- $\widehat{s}_j$ is an estimate of the true signal s_j (e.g., c_j from the centroid),
- $\lambda > 0$ balances the trade-off between the two terms,
- $m_j \in \{0,1\}$ enforces a hard-threshold selection of dimensions.

Minimizing Eq. 2 encourages the algorithm to keep dimensions that best align with $\mathbf{q}$ (where $q_j\,s_j$ is large) and filter out dimensions that are mostly noisy (large variance η_j with little or negative contribution to the query).

REMARK 1. *The actual, true, signal* **s** *cannot be directly observed. In practice, one may replace* $\widehat{s_j}$ *with* c_j*, the* j*-th component of the centroid, or use another form of aggregation (e.g., a average over the top-*k *documents). This makes it possible to implement the objective in Eq. 2 without direct access to the true signal vector.*

Signal Estimation Since s_j is unobserved, the query-centroid score is used as an estimator:

$$\widehat{q_j s_j} = q_j c_j = r_j(q_j, c_j)$$

Dimension Importance Estimation. The original DIME paper uses the following heuristic to select the m mask.

$$m_j^* = \begin{cases} 1 & \text{if } q_j c_j \text{ is among top-}l\text{ values} \\ 0 & \text{otherwise} \end{cases}$$

where $0 < l < h$ is an hyperparameter.

Bounding Noise via Chebyshev's Inequality. We use the Chebyshev inequality to provide a probabilistic guarantee for the deviation of a random variable from its mean, without requiring any specific distribution (e.g., Gaussian). For the noise term $\eta_j = \frac{1}{k}\sum_{i=1}^{k} \epsilon_j^{(i)}$, which has mean zero and variance $\mathrm{Var}(\eta_j) = \frac{\sigma^2}{k}$, Chebyshev's inequality states:

$$\mathbb{P}\left(|\eta_j| \geq t \cdot \sqrt{\mathrm{Var}(\eta_j)}\right) = \mathbb{P}\left(|\eta_j| \geq t \cdot \frac{\sigma}{\sqrt{k}}\right) \leq \frac{1}{t^2},$$

where $t > 0$ is a scaling factor. This inequality bounds the probability that the noise term η_j exceeds a threshold proportional to its standard deviation. For example, setting $t = \sqrt{k}$ yields:

$$\mathbb{P}\left(|\eta_j| \geq \sigma\right) \leq \frac{1}{k}.$$

This means that with probability at least $1 - \frac{1}{k}$, the noise term η_j is bounded by σ_j. As k (the number of pseudo-relevant documents) increases, this bound becomes tighter, ensuring that η_j remains small. With error bound:

$$|r_j(q_j, c_j) - q_j s_j| = |q_j \eta_j| \leq |q_j| \sqrt{\frac{\sigma^2}{k}} \quad \text{(w.h.p. via Chebyshev)}$$

3.2 Improved Weighted Average

The original DIME approach aggregates the top-k document embeddings by a simple (uniform) average, which gives an effective signal strength of

$$\bar{\alpha}_{\text{uniform}} = \frac{1}{k}\sum_{i=1}^{k} \alpha^{(i)}.$$

In contrast, our improved approach uses a weighted average:

$$\bar{\alpha}_{\text{weighted}} = \sum_{i=1}^{k} w^{(i)}\, \alpha^{(i)},$$

where the weights $w^{(i)} \in [0, 1]$ are computed from the query-document similarity scores. For example, we may set

$$w^{(i)} = \sigma(\bar{r})^{(i)} \;\mid\; \bar{r} = \left\{ r(q, d^{(i)}) \right\}_{i=1}^{k},\; \sigma(z)^{(i)} = \frac{e^{z_i/\tau}}{\sum_{\ell=1}^{k} e^{z_\ell/\tau}} \qquad (3)$$

where σ is the softmax function with a temperature parameter $\tau > 0$ that controls the sharpness of the weight distribution. Under the assumption that documents with higher relevance scores tend to exhibit a stronger expression of the core information need (i.e., larger $\alpha^{(i)}$), it is reasonable to expect that

$$\bar{\alpha}_{\text{weighted}} > \bar{\alpha}_{\text{uniform}},$$

thereby emphasizing the more informative documents in the aggregated signal.

Noise Analysis of the Weighted Average

Following Eq. 1, the aggregated noise term for the weighted centroid becomes

$$\eta_{w,j} = \sum_{i=1}^{k} w^{(i)}\, \epsilon_j^{(i)}.$$

Because the weights $w^{(i)}$ depend on the document scores (and hence potentially on the noise), we use the law of total expectation to verify that the weighted noise remains unbiased. Specifically,

$$\begin{aligned} \mathbb{E}[\eta_{w,j}] &= \mathbb{E}\left[\sum_{i=1}^{k} w^{(i)}\, \epsilon_j^{(i)}\right] = \sum_{i=1}^{k} \mathbb{E}\left[w^{(i)}\, \epsilon_j^{(i)}\right] \\ &= \sum_{i=1}^{k} \mathbb{E}\left[\mathbb{E}\left[w^{(i)}\, \epsilon_j^{(i)} \mid w^{(i)}\right]\right] \quad \text{(by law of total expectation)} \\ &= \sum_{i=1}^{k} \mathbb{E}\left[w^{(i)}\, \mathbb{E}\left[\epsilon_j^{(i)} \mid w^{(i)}\right]\right] \\ &= \sum_{i=1}^{k} \mathbb{E}\left[w^{(i)} \cdot 0\right] = 0. \quad (\text{since } \mathbb{E}[\epsilon_j^{(i)} \mid w^{(i)}] = 0) \end{aligned}$$

Thus, even if $w^{(i)}$ and $\epsilon_j^{(i)}$ are dependent, the expected value of the aggregated noise is zero, and the variance is

$$\mathrm{Var}(\eta_{w,j}) = \sigma^2 \sum_{i=1}^{k} \left(w^{(i)}\right)^2.$$

Signal-to-Noise Ratio comparison Via Chebyshev's inequality, the error bound for the uniform and weighted averages are

$$|q_j \eta_j| \lesssim |q_j| \sqrt{\frac{\sigma^2}{k}}, \qquad |q_j \eta_{w,j}| \lesssim |q_j| \sqrt{\sigma^2 \sum_{i=1}^{k} (w^{(i)})^2}.$$

We can define the per-dimension SNR as:

$$\mathrm{SNR}_{\text{uniform}} = \frac{|q_j\, s_j|\; \bar{\alpha}_{\text{uniform}}}{\sigma/\sqrt{k}}, \qquad \mathrm{SNR}_{\text{weighted}} = \frac{|q_j\, s_j|\; \bar{\alpha}_{\text{weighted}}}{\sigma\sqrt{\sum_{i=1}^{k} \left(w^{(i)}\right)^2}}.$$

Even if the weighted average introduces a slightly larger noise variance (since $\sqrt{\sum_{i=1}^{k} \left(w^{(i)}\right)^2} \geq \frac{1}{\sqrt{k}}$), the improvement in the effective signal $\bar{\alpha}_{\text{weighted}}$ often more than compensates. In particular, if

$$\frac{\bar{\alpha}_{\text{weighted}}}{\sqrt{\sum_{i=1}^{k} \left(w^{(i)}\right)^2}} > \bar{\alpha}_{\text{uniform}} \sqrt{k},$$

then we obtain

$$\mathrm{SNR}_{\text{weighted}} > \mathrm{SNR}_{\text{uniform}}.$$

For instance, let's assume that the signal strength is proportional to the weight (as it is, in turn, proportional to the score), i.e.,

$$\alpha^{(i)} = c\, w^{(i)} \quad \text{with } c > 0.$$

Then,

$$\bar{\alpha}_{\text{uniform}} = \frac{c}{k}, \qquad \bar{\alpha}_{\text{weighted}} = c \sum_{i=1}^{k} \left(w^{(i)} \right)^2,$$

since $\sum_{i=1}^{k} w^{(i)} = 1$. The SNR condition then reduces to

$$\frac{c \sum_{i=1}^{k} \left(w^{(i)} \right)^2}{\sqrt{\sum_{i=1}^{k} \left(w^{(i)} \right)^2}} > \frac{c}{\sqrt{k}} \Rightarrow \sqrt{\sum_{i=1}^{k} \left(w^{(i)} \right)^2} > \frac{1}{\sqrt{k}} \Rightarrow \sum_{i=1}^{k} \left(w^{(i)} \right)^2 > \frac{1}{k}.$$

This inequality is strict whenever the weights are not uniformly distributed—that is, when the aggregation concentrates more on the top documents. To ensure this, we use a low τ (temperature) in the softmax (Eq. 3), in fact, as τ decreases, the weight distribution becomes more concentrated, which can further improve the SNR if the top documents are significantly more relevant.

4 DIME reproduction

We begin by reproducing the core results of DIME, specifically focusing on two variants: PRF DIME and LLM DIME. To ensure fidelity to the original study, we adopt the same datasets, models, parameters, and overall configuration described by the original authors. These details–along with the precise experimental setup–are presented in the following section. By closely mirroring the original design, the aim is to validate our implementation and confirm that the reproduced results align with those initially reported.

4.1 Experimental Setup

In this section, we detail the setup used to reproduce the main results of DIME, focusing on the original *embedding models*, *datasets*, *retrieval algorithms*, and *evaluation metrics*.

Embedding Models. We investigate the same dense encoders:

- **ANCE** [42] uses hard-negative mining during contrastive learning to improve robustness.
- **Contriever** [14] emphasizes unsupervised contrastive pretraining, aiming for strong zero-shot generalization.
- **TAS-B** [13] employs distillation-based training with topic-aware sampling.

All models produce 768-dimensional embeddings (unless otherwise stated) and are optimized for dot-product.

Datasets. MS Marco passage[4], We use two primary datasets:

- **MS MARCO Passage** [25] is a large-scale passage retrieval benchmark containing real-world queries from Bing's search logs. This collection is commonly used to fine-tune and evaluate dense retrieval methods. We also utilize the TREC Deep Learning (DL) tracks 2019 and 2020, as well as the Hard subset, to gauge performance in more challenging query scenarios.
- **TREC Robust** (a.k.a. Robust '04) uses documents from TREC disks 4 and 5 [36, 37], minus the Congressional Record. The Robust retrieval track was designed to emphasize "difficult" or poorly performing topics, thereby testing a system's consistency and resilience.

Implementation. We index the document embeddings using the `Flat` index within the FAISS library, which stores all vectors in raw form (i.e., without compression) and supports efficient exhaustive nearest neighbor searches. By default, we retrieve the top k candidate passages per query; unless otherwise specified, $k = 10$. Distances or similarities between the query and document embeddings are computed by an inner product. We reproduced both DIME's approaches:

- The pseudo-relevance feedback variant, $\{u_{\text{PRF@}k}$, by adjusting the query vector (or its dimensions) via centroid-based feedback from the top k retrieved documents.
- We also investigate the LLM-based estimator (u_{LLM}), which relies on synthetic passages generated by a language model for dimensional scoring.

Relatively to the latter, we perform LLM-based answer generation of each of the queries evaluated using the official GPT-4 API, setting the generation seed as provided in the reference implementation.

Measures. We evaluate retrieval performance using the standard nDCG@10 metric, which places emphasis on ranking high-quality documents near the top of the results list. In particular, we report nDCG@10 across various queries from each collection to assess consistency and effectiveness in both in-domain (MS MARCO) and out-of-domain (TREC Robust) scenarios.

Hardware. All experiments are conducted on a computational cluster equipped with eight NVIDIA A100 GPUs. Each job runs on a single GPU unless otherwise specified. Our implementation is primarily in Python, making use of PyTorch for model loading and FAISS for efficient similarity search.

4.2 Results

Table 1 reports nDCG@10 scores for our main reproduction of DIME's original results, as well as a comparison to the originally reported figures. We present outcomes on four test sets—*DL '19*, *DL '20*, *DL Hard*, and *Robust '04*—and group results by encoder (ANCE, Contriever, TAS-B). Within each group, we show performance for each DIME instantiation ($u_{\text{PRF@1}}, u_{\text{PRF@2}}, u_{\text{PRF@5}}$, and u_{LLM}) across varying proportions of retained dimensions (e.g., 0.2, 0.4, 0.6, 0.8, 1).

Overall, the reproduced results closely align with those initially reported, demonstrating the stability of DIME under different experimental conditions. Specifically:

- **PRF-based DIMEs** ($u_{\text{PRF@}k}$): These consistently improve over the no-filtering baseline in most settings, with the largest gains typically observed on the TREC Robust collection, suggesting that *pseudo-relevance feedback* can be particularly beneficial for difficult queries.
- **LLM-based DIME** (u^{LLM}): While computationally more expensive (due to the generation of synthetic passages), it often yields the best or near-best performance among the DIME variants. Although relying on a third-party API-based model to generate the synthetic documents, we were able to reproduce the improvements.
- **Encoder Differences**: ANCE tends to benefit moderately from dimension filtering; Contriever and TAS-B show especially strong improvements on collections like DL Hard and Robust '04, potentially thanks to how their pre-training

Table 1: Comparison of nDCG@10 performance across TREC Deep Learning 2019, 2020, Hard, and Robust 2004 datasets, showing DIME reproduction and original DIME results for ANCE, Contriever, and TAS-B under varying parameter settings.

		DL '19					DL '20					DL HD					RB '04				
		0.2	0.4	0.6	0.8	1	0.2	0.4	0.6	0.8	1	0.2	0.4	0.6	0.8	1	0.2	0.4	0.6	0.8	1
DIME Reproduction Results																					
ANCE	$u^{PRF@1}$	.082	.553	.638	.650	.643	.176	.539	.612	.644	.644	.055	.272	.331	.329	.326	.061	.259	.312	.330	.327
	$u^{PRF@2}$	.089	.565	.637	.649		.166	.543	.612	.643		.046	.268	.323	.331		.059	.251	.312	.325	
	$u^{PRF@5}$	.085	.563	.635	.648		.159	.543	.614	.644		.050	.276	.335	.332		.056	.247	.311	.329	
	u^{LLM}	.091	.548	.660	.663		.149	.544	.634	.651		.033	.291	.340	.344		.065	.258	.332	.337	
Contriever	$u^{PRF@1}$	.675	.683	.686	.689	.674	.711	.704	.702	.693	.672	.388	.386	.388	.390	.377	.484	.493	.495	.494	.466
	$u^{PRF@2}$	.673	.675	.679	.685		.683	.685	.685	.685		.394	.395	.387	.389		.493	.497	.498	.494	
	$u^{PRF@5}$	.649	.664	.678	.683		.695	.686	.690	.685		.377	.378	.387	.387		.477	.479	.480	.476	
	u^{LLM}	.729	.741	.751	.749		.723	.720	.714	.712		.408	.406	.414	.405		.497	.505	.499	.498	
TAS-B	$u^{PRF@1}$	.722	.732	.734	.729	.718	.700	.702	.711	.705	.684	.359	.374	.383	.375	.376	.433	.449	.450	.455	.447
	$u^{PRF@2}$	.719	.731	.731	.725		.687	.698	.709	.707		.352	.373	.373	.374		.460	.469	.467	.467	
	$u^{PRF@5}$	.712	.725	.721	.724		.686	.685	.694	.697		.369	.385	.389	.395		.462	.467	.469	.469	
	u^{LLM}	.757	.755	.757	.759		.708	.707	.717	.718		.389	.413	.418	.405		.456	.471	.476	.475	
Original DIME Results																					
ANCE	$u^{PRF@1}$	.082	.559	.644	.658	.643	.175	.549	.616	.648	.644	.042	.266	.326	.332	.325	.074	.284	.343	.357	.362
	$u^{PRF@2}$	.095	.567	.637	.652		.176	.542	.612	.647		.051	.274	.325	.328		.066	.273	.341	.356	
	$u^{PRF@5}$	.088	.568	.633	.647		.155	.545	.613	.645		.054	.274	.330	.330		.058	.263	.334	.359	
	u^{LLM}	.081	.569	.651	.663		.171	.537	.629	.655		.042	.284	.339	.348		.078	.280	.354	.371	
Contriever	$u^{PRF@1}$	.676	.685	.686	.689	.675	.711	.703	.701	.692	.672	.396	.395	.387	.389	.377	.512	.522	.527	.523	.499
	$u^{PRF@2}$	.672	.675	.679	.685		.682	.685	.687	.685		.395	.391	.394	.399		.500	.513	.517	.515	
	$u^{PRF@5}$	.646	.664	.680	.681		.698	.687	.690	.686		.379	.385	.383	.387		.504	.513	.511	.512	
	u^{LLM}	.720	.742	.752	.750		.719	.722	.725	.710		.392	.409	.414	.412		.527	.539	.539	.530	
TAS-B	$u^{PRF@1}$	.719	.731	.733	.729	.718	.697	.699	.709	.703	.684	.349	.376	.374	.375	.376	.458	.475	.475	.471	.453
	$u^{PRF@2}$	.718	.733	.731	.726		.684	.698	.710	.707		.359	.377	.382	.391		.465	.474	.476	.470	
	$u^{PRF@5}$	.709	.721	.719	.721		.683	.687	.693	.695		.364	.371	.384	.381		.462	.460	.462	.464	
	u^{LLM}	.747	.749	.760	.755		.708	.706	.710	.712		.385	.397	.401	.397		.462	.487	.488	.485	

strategy helps handle out-of-distribution queries and have a better separation among topics in their latent space.

- **Robust '04**: For this dataset, our experiments show slightly lower retrieval quality for both the baseline as well as the results with fewer retained dimensions. What is important to notice though is that the reducing dimensions improves the baseline in the same way as reported by the original authors, which indicates that the offset could derive from minor differences in the settings used.

In summary, Table 1 affirms that DIME approaches can significantly bolster dense retrieval effectiveness, particularly in challenging or out-of-domain settings, and that our replication setup matches the trends reported in the original study.

5 Generalization Study

In this section, we expand the scope of DIME's evaluation to investigate its broader applicability beyond the original settings. Specifically, we test models trained with different objectives (e.g., cosine similarity), explore embeddings that follow a *Matryoshka* structure, and examine higher-dimensional representations. We further evaluate DIME on new datasets—including zero-shot scenarios—and experiment with various LLMs beyond GPT-4. By diversifying both the model and dataset landscapes, we aim to determine whether DIME's reported benefits remain robust across a range of architectures and configurations, thereby shedding light on its potential for broader deployment.

5.1 Experimental Setup

Embedding Models. To broaden the scope of our findings, we test:

- **Multilingual E5**[1] [38, 39], e5-large in this paper, handles cross-lingual queries and documents for multilingual retrieval scenarios.
- **Mxbai-Embed Large**[2], mxbai in this paper, targets robust, versatile embeddings for downstream tasks.
- **Snowflake-Arctic Embed**[3] [46], arctic-v2 in this paper, leverages an advanced training curriculum to produce strong general-purpose embeddings.
- **BAAI-BGE-M3**[4] [24, 41], bge-m3 in this paper, combines balanced representations for both short and long passages, demonstrating robustness across diverse benchmarks.

LLMs. We evaluate LLM DIME (described in Section 2) across a variety of LLMs:

- **GPT-4** and **GPT-4o** [1, 15], OpenAI's state-of-the-art models, are the version used in the original DIME framework and its latest release, respectively;
- **Qwen 2.5** [44], an open source multilingual general purpose family of LLMs showcasing performance comparable to proprietary models, which we use in its 32B, 7B and 3B parameters variants;

[1] https://huggingface.co/intfloat/multilingual-e5-large
[2] https://huggingface.co/mixedbread-ai/mxbai-embed-large-v1
[3] https://huggingface.co/Snowflake/snowflake-arctic-embed-l-v2.0
[4] https://huggingface.co/BAAI/bge-m3

- **Llama 3.1** and **3.2** [9], similarly to Qwen, a family of open source multilingual models, which we use in its 8B and 3B parameters variants;
- **DeepSeek R1** [12], a family of models trained for improved reasoning capabilities; we use the distilled 32B and 7B parameters Qwen-based variants.

For all models, `temperature` is set to 0.5 and `top_p` is set to 0.7 during generation. For models with reasoning capabilities the thinking tokens have been removed from the generated answer.

Datasets. We extended our evaluation to the zero-shot BEIR benchmark [32], excluding the four corpora that are not publicly available to ensure reproducibility.

Hardware. For the generalization study, we build on the experimental setup used to replicate the results detailed in Section 4.1. Inference on open source LLMs is performed using vLLM [18], with 32B models quantized to 4 bits to efficiently fit within the memory constraints of a single GPU.

5.2 Generalization to other models

To verify whether our Dimension Importance Estimation (DIME) strategy applies beyond the specific encoders described in earlier sections, we extend our analysis to a range of other dense retrieval models, as summarized in Table 2. Each model brings distinct training objectives or domain focuses; for example, bge-m3 aims at broader domain coverage, mxbai targets multilingual or multi-domain robustness, e5-large emphasizes multilingual embeddings, and arctic-v2 seeks strong general-purpose representations through advanced curriculum design.

Despite these architectural and training variations, the table reveals several consistent patterns when pseudo-relevance feedback (PRF) is used to guide dimension filtering. First, all models generally achieve noticeable improvements in nDCG@10 by discarding a subset of dimensions, mirroring earlier trends observed with ANCE, Contriever, and TAS-B. Second, the exact fraction of coordinates retained (e.g., 40–60%) can slightly shift among models, yet in every case there appears to be a sweet spot that meaningfully outperforms using the full set of latent dimensions. Third, models tested in out-of-domain or more challenging settings—such as TREC Robust '04 or the DL HD subset—continue to benefit from dimension filtering, suggesting that DIME's focus on query-relevant coordinates helps counteract domain mismatches or inherently difficult topics.

Overall, these additional experiments support the conclusion that DIME is broadly applicable across diverse embedding architectures. Whether the encoder is multilingual, specialized for certain domains, or generically trained, adaptively pruning dimensions appears to reduce noise and sharpen the representation, thus enhancing retrieval performance in both familiar and zero-shot scenarios.

In contrast to the models tested in the original work, all models adopted in this section, with the exception of BAAI-BGE-M3, are trained to optimize cosine similarity. In order to leverage the dot-product as a signal estimation mechanism, the produced vectors have been normalized. It is interesting to note how dimension masking, which affects vector normalization, does not cause harm to the retrieval quality. This is due to the way DIME performs dimension importance estimation, dropping the dimensions with a low score.

5.3 Generalization to Zero-Shot Datasets

While earlier experiments focused on in-domain and closely related benchmarks, many real-world applications involve queries and content that deviate from a model's training distribution. We therefore assess whether DIME can enhance retrieval in *zero-shot* scenarios. In particular, we use the BEIR benchmark [32], which comprises diverse IR tasks such as fact verification (Fever), argumentative retrieval (ArguAna), and financial QA (FiQA).

Table 3 reports nDCG@10 scores across 13 BEIR tasks. Each row corresponds to a dataset and pseudo-relevance feedback variant ($u^{PRF@1}$, $u^{PRF@2}$, or $u^{PRF@5}$), while columns indicate the fraction of dimensions retained (0.2, 0.4, 0.6, 0.8) for each model. Several consistent trends emerge:

- **Cross-Dataset Robustness.** Even though the encoders were trained on unrelated data, dimension pruning often boosts performance. For instance, on `c-fever`, TAS-B sees noticeable gains when moving from 0.8 to 0.2 retained dimensions under $u^{PRF@1}$.
- **Varying Optimal Prune Rates.** Different tasks favor different levels of pruning. On `arguana`, TAS-B achieves its highest nDCG@10 at around 40% or 60% dimension retention, whereas on `fiqa`, a more aggressive prune rate (e.g., 20%) can yield stronger gains with $u^{PRF@2}$ or $u^{PRF@5}$.
- **Minimal Regression on Strong Baselines.** Tasks such as `quora` and `nq` show only marginal changes because both models already perform strongly out-of-the-box. In these cases, dimension pruning neither substantially helps nor harms, indicating a relatively safe application of DIME.

These results suggest that DIME's benefits generalize beyond the domain of its original training data. In particular, by focusing on the coordinates that most strongly align with the pseudo-relevant feedback, DIME can reduce noisy dimensions even when the query or document distribution shifts dramatically. This finding is particularly valuable for applications that must handle diverse tasks or emerging topics without specialized training or fine-tuning.

5.4 LLM-Based Filtering with Different Models

Table 4 reports nDCG@10 scores for various Large Language Models (LLMs) used to generate synthetic text for LLM-based DIME. The experiments span two encoders (TAS-B and arctic-v2) over three TREC Deep Learning collections (DL '19, DL '20, DL HD), with each row representing a distinct LLM. Our selection covers high-parameter, proprietary systems (GPT-4, GPT-4o) as well as mid-scale open-source alternatives (Qwen 2.5, Llama 3.x) and an advanced reasoning models (Distilled DeepSeek R1).

Across collections, most LLMs yield competitive results. GPT-4 and GPT-4o generally rank at or near the top, although the optimal fraction of retained dimensions varies with the dataset and encoder. For instance, GPT-4 with TAS-B attains peak performance on DL '20 at 0.6 or 0.8 retained dimensions, while on DL '19 the gains saturate around 0.4–0.6. This variability underscores that the ideal level of dimension pruning depends on both the underlying encoder and the query distribution.

In general, relatively small open-source models such as Llama-3.1_{8B} can be used as an open and cheaper alternative to GPT-4. When queries are harder—as in DL HD—larger (but still open)

Table 2: nDCG@10 performance for the bge-m3, mxbai, e5-large, and arctic-v2 retrieval models, showing comparisons across TREC Deep Learning 2019, 2020, Hard, and Robust 2004 under different under varying parameter settings.

		DL '19					DL '20					DL HD					RB '04				
		0.2	0.4	0.6	0.8	1	0.2	0.4	0.6	0.8	1	0.2	0.4	0.6	0.8	1	0.2	0.4	0.6	0.8	1
bge-m3	$u^{PRF@1}$	.700	.700	.699	.696	.668	.692	.708	.711	.706	.677	0.358	.359	.365	.362	.335	.427	.443	.444	.444	.424
	$u^{PRF@2}$	.687	.689	.694	.690		.705	.708	.706	.701		0.354	.359	.358	.363		.434	.443	.442	.442	
	$u^{PRF@5}$	.696	.700	.702	.688		.677	.680	.685	.685		.343	.358	.360	.362		.428	.430	.433	.435	
	u^{LLM}	.749	.737	.733	.729		.739	.746	.739	.733		.381	.396	.384	.379		.451	.465	.465	.466	
mxbai	$u^{PRF@1}$	.714	.726	.722	.722	.694	.695	.710	.712	.709	.705	.376	.391	.393	.390	.381	.477	.479	.489	.492	.483
	$u^{PRF@2}$	.722	.724	.722	.718		.710	.717	.722	.723		.386	.401	.407	.405		.486	.493	.497	.496	
	$u^{PRF@5}$	.706	.716	.718	.715		.727	.732	.721	.717		.393	.403	.404	.402		.488	.489	.492	.492	
	u^{LLM}	.752	.763	.749	.737		.732	.734	.735	.729		.407	.407	.401	.400		.486	.493	.490	.484	
e5-large	$u^{PRF@1}$	.721	.748	.744	.747	.719	.701	.730	.744	.740	.723	.385	.384	.397	.402	.379	.440	.459	.470	.469	.450
	$u^{PRF@2}$	.719	.740	.739	.734		.712	.741	.741	.730		.369	.374	.393	.394		.455	.465	.476	.471	
	$u^{PRF@5}$	.717	.733	.735	.732		.710	.727	.728	.730		.368	.374	.385	.375		.461	.472	.481	.468	
	u^{LLM}	.731	.760	.748	.746		.711	.735	.738	.735		.375	.386	.398	.398		.442	.474	.482	.478	
arctic-v2	$u^{PRF@1}$	.740	.742	.737	.739	.725	.735	.743	.746	.747	.733	.406	.407	.406	.408	.381	.492	.499	.500	.500	.466
	$u^{PRF@2}$	.740	.739	.736	.741		.732	.741	.744	.746		.368	.399	.398	.400		.499	.503	.501	.497	
	$u^{PRF@5}$	.739	.743	.735	.734		.739	.739	.744	.746		.399	.399	.400	.395		.491	.492	.494	.494	
	u^{LLM}	.750	.745	.742	.736		.711	.715	.719	.708		.409	.410	.405	.395		.470	.477	.477	.478	

models like Qwen2.5$_{32B}$ tend to yield better results, suggesting that GPT-4 is not necessary overall. Although larger models (e.g., GPT-4, Qwen2.5$_{32B}$, Llama-3.1$_{8B}$) typically achieve slightly higher metrics than their smaller counterparts (e.g., Qwen2.5$_{3B}$, Llama-3.2$_{3B}$), architectural efficiency or domain adaptation can sometimes compensate for differences in parameter count.

With regard to DeepSeek R1, it produces shorter responses that contain less semantic content useful for discriminating among dimensions, which leads to slightly lower performance compared to other models.

Finally, while GPT-4 frequently delivers the highest effectiveness, its higher latency and computational costs suggest that mid-scale open-source models offer a favorable trade-off between retrieval gains and inference speed.

6 Improvements

In this section, we present the proposed improvements to DIME.

6.1 Softmax-Weighted Centroid (SWC)

In the original DIME formulation, the centroid of the top-k pseudo-relevant documents is computed as the simple arithmetic mean of their embedding vectors. This centroid is then used to estimate the importance of each dimension (cf. Section 2). However, in Section 3.2 we show that the presence of varying relevance scores may suggest that not all documents should contribute equally. To address this, we propose a new approach called *Softmax-Weighted Centroid* (SWC).

Instead of averaging the top-k document representations with uniform weights, we leverage a temperature-scaled softmax function to assign higher weights to more relevant matches. Specifically, let $\mathcal{M}$ be the set of top-ranked matches for a given query, each match $m \in \mathcal{M}$ comprising an embedding $\mathbf{v}_m$ and a scalar relevance score r_m. Denoting $\mathbf{V}$ as the collection of embeddings and $\mathbf{r} = \{r_m\}_{m=1}^{k}$ as the corresponding scores, we compute the centroid as $\mathbf{c} = \sum_{m=1}^{k} \left(\alpha_m \, \mathbf{v}_m \right)$ where $\alpha_m = \sigma(\mathbf{r})_m$. We use a low τ (temperature of the softmax, defined in Eq. 3), in fact, as τ decreases, the weight distribution becomes sharper, which can further improve the SNR if the top documents are significantly more relevant.

Table 5 key properties:

- **Score-Aware Averaging:** Unlike uniform averaging, SWC raises the contribution of documents with higher retrieval scores, making the final centroid more representative of strongly relevant matches.
- **Temperature-Scaling:** The temperature $\tau > 0$ controls how sharply the weights concentrate on high-scoring documents. A small τ emphasizes top-scoring matches, while a large τ yields a nearly uniform distribution.
- **Flexibility and Compatibility:** The SWC approach can be easily integrated with any dimension-importance estimator that uses a centroid (e.g., PRF DIME), by simply replacing the arithmetic mean with the score-weighted centroid.

6.2 DIME-based Reranking

Table 6 illustrates how DIME can be leveraged as a score refinement technique applied directly to the top 100 documents retrieved by the initial query. This approach is particularly justified by the first query's high recall, which ensures that the most relevant documents are already present in the initial retrieval set. By refining scores within this restricted subset, DIME effectively enhances ranking quality while maintaining computational efficiency, as it avoids the need for a second query which might increase latency. This shows the practical applicability of the method in real-world retrieval pipelines, where balancing effectiveness and efficiency is crucial.

7 Conclusion

In this work, we provided a comprehensive study of Dimension IMportance Estimation (DIME), confirming its effectiveness for dense retrieval through both replication of prior experiments and extensive new evaluations. Our results show that selectively pruning

Table 3: nDCG@10 performance on the BEIR benchmark.

		TAS-B					arctic-v2				
		0.2	0.4	0.6	0.8	1	0.2	0.4	0.6	0.8	1
arguana	$u^{PRF@1}$	.366	.361	.351	.340		.442	.441	.440	.438	
	$u^{PRF@2}$	.332	.333	.331	.331	.321	.428	.429	.430	.431	.431
	$u^{PRF@5}$	.319	.319	.320	.322		.420	.423	.424	.426	
c-fever	$u^{PRF@1}$	.240	.241	.240	.241		.368	.381	.387	.394	
	$u^{PRF@2}$	.244	.239	.238	.237	.228	.386	.396	.401	.405	.418
	$u^{PRF@5}$	.252	.246	.244	.242		.398	.407	.410	.413	
dbpedia	$u^{PRF@1}$	.368	.375	.376	.379		.416	.425	.431	.434	
	$u^{PRF@2}$	.377	.380	.379	.380	.384	.425	.432	.435	.436	.434
	$u^{PRF@5}$	.377	.379	.377	.382		.428	.433	.439	.438	
fever	$u^{PRF@1}$	.665	.686	.694	.697		.910	.912	.913	.914	
	$u^{PRF@2}$	.686	.712	.717	.717	.700	.909	.914	.914	.915	.915
	$u^{PRF@5}$	.682	.720	.727	.725		.901	.908	.910	.913	
fiqa	$u^{PRF@1}$	.279	.286	.293	.295		.441	.446	.450	.452	
	$u^{PRF@2}$	.281	.283	.287	.288	.300	.440	.448	.448	.449	.454
	$u^{PRF@5}$	.273	.275	.281	.289		.428	.437	.438	.440	
hotpotqa	$u^{PRF@1}$	.525	.540	.548	.555		.636	.647	.653	.658	
	$u^{PRF@2}$	.550	.557	.563	.567	.584	.657	.663	.666	.668	.682
	$u^{PRF@5}$	.525	.547	.555	.561		.650	.659	.665	.667	
nfcorpus	$u^{PRF@1}$	.318	.325	.327	.327		.364	.369	.368	.368	
	$u^{PRF@2}$	.321	.317	.323	.325	.319	.367	.367	.369	.366	.352
	$u^{PRF@5}$	.308	.316	.321	.320		.357	.360	.363	.361	
nq	$u^{PRF@1}$	.445	.453	.458	.461		.619	.625	.628	.630	
	$u^{PRF@2}$	.453	.457	.457	.460	.463	.617	.627	.628	.630	.637
	$u^{PRF@5}$	.449	.452	.454	.458		.612	.620	.626	.631	
quora	$u^{PRF@1}$	.824	.825	.826	.828		.883	.885	.886	.886	
	$u^{PRF@2}$	.822	.821	.823	.826	.835	.884	.886	.887	.887	.888
	$u^{PRF@5}$	.801	.804	.811	.818		.871	.878	.881	.883	
scidocs	$u^{PRF@1}$	.156	.151	.151	.152		.204	.206	.206	.207	
	$u^{PRF@2}$	.152	.147	.146	.146	.149	.205	.208	.207	.208	.203
	$u^{PRF@5}$	.148	.143	.141	.142		.201	.206	.206	.206	
scifact	$u^{PRF@1}$	.619	.624	.626	.627		.692	.694	.699	.701	
	$u^{PRF@2}$	.622	.621	.625	.626	.643	.703	.712	.707	.710	.710
	$u^{PRF@5}$	.597	.610	.612	.617		.683	.700	.696	.701	
w-touche	$u^{PRF@1}$	.163	.166	.161	.166		.249	.251	.252	.253	
	$u^{PRF@2}$	.154	.168	.167	.165	.163	.243	.249	.256	.254	.259
	$u^{PRF@5}$	.161	.163	.165	.162		.250	.252	.253	.257	
t-covid	$u^{PRF@1}$	.434	.467	.478	.498		.826	.833	.831	.833	
	$u^{PRF@2}$	.456	.489	.501	.507	.481	.829	.831	.835	.843	.834
	$u^{PRF@5}$	.484	.503	.510	.511		.825	.839	.838	.843	

Table 4: nDCG@10 scores for various LLMs (GPT-4, GPT-4o, Qwen, Llama, and DeepSeek) used to generate synthetic text for LLM-based DIME, evaluated on TREC Deep Learning 2019, 2020, and Hard tasks with TAS-B and arctic-v2 retrievers.

		DL '19				DL '20				DL HD			
		0.2	0.4	0.6	0.8	0.2	0.4	0.6	0.8	0.2	0.4	0.6	0.8
TAS-B	GPT-4	.757	.755	.757	.759	.708	.707	.717	.718	.389	.413	.418	.405
	GPT-4o	.749	.758	.756	.754	.703	.699	.700	.705	.393	.393	.391	.392
	Qwen2$_{3B}$	.698	.733	.744	.741	.687	.689	.696	.694	.381	.394	.401	.400
	Qwen2$_{7B}$	.717	.730	.745	.743	.695	.691	.692	.695	.337	.363	.381	.386
	Qwen2$_{32B}$	.743	.756	.754	.749	.694	.702	.708	.705	.398	.390	.401	.399
	Llama-3.2$_{3B}$	.712	.726	.731	.731	.692	.695	.703	.697	.358	.358	.371	.378
	Llama-3.1$_{8B}$	.750	.756	.758	.750	.695	.711	.708	.702	.375	.383	.392	.388
	DeepSeek$_{32B}$	.710	.726	.730	.734	.677	.695	.702	.699	.377	.387	.393	.396
	DeepSeek$_{7B}$	.687	.712	.713	.718	.659	.677	.687	.688	.361	.372	.374	.384
arctic-v2	GPT-4	.750	.745	.742	.736	.711	.715	.719	.708	.409	.410	.405	.395
	GPT-4o	.735	.746	.735	.725	.703	.707	.704	.701	.390	.390	.392	.386
	Qwen2$_{3B}$	.723	.723	.721	.723	.675	.684	.680	.679	.391	.394	.389	.386
	Qwen2$_{7B}$	.728	.734	.728	.722	.701	.701	.707	.701	.361	.382	.380	.377
	Qwen2$_{32B}$	.743	.741	.731	.726	.702	.696	.697	.692	.393	.396	.384	.383
	Llama-3.2$_{3B}$	.721	.724	.712	.706	.689	.698	.697	.690	.375	.372	.368	.371
	Llama-3.1$_{8B}$	.746	.750	.738	.726	.711	.708	.699	.692	.404	.395	.390	.389
	DeepSeek$_{32B}$	.722	.727	.723	.722	.703	.695	.693	.689	.386	.393	.386	.383
	DeepSeek$_{7B}$	.670	.681	.677	.670	.654	.659	.668	.652	.362	.368	.374	.374

Table 5: nDCG@10 performance on TREC Deep Learning 2019, 2020, and Hard tasks using Softmax-Weighted average.

	DL '19				DL '20				DL HD			
	0.2	0.4	0.6	0.8	0.2	0.4	0.6	0.8	0.2	0.4	0.6	0.8
						$u^{PRF@10}$						
TAS-B	.709	.711	.721	.720	.676	.684	.686	.693	.352	.370	.377	.378
arctic-v2	.711	.725	.727	.726	.725	.729	.737	.736	.393	.390	.398	.395
						$SWC^{PRF@10}$						
TAS-B	.725	.733	.732	.727	.708	.710	.718	.713	.370	.384	.386	.380
arctic-v2	.726	.738	.727	.729	.731	.734	.737	.737	.399	.399	.400	.395

Table 6: nDCG@10 performance on TREC Deep Learning 2019, 2020, and Hard tasks when DIME is used as a reranker.

	DL '19				DL '20				DL HD			
	0.2	0.4	0.6	0.8	0.2	0.4	0.6	0.8	0.2	0.4	0.6	0.8
						Original DIME						
TAS-B	.712	.725	.721	.724	.686	.685	.694	.697	.369	.385	.389	.395
arctic-v2	.739	.743	.735	.734	.739	.739	.744	.746	.399	.399	.400	.395
						Reranking						
TAS-B	.712	.725	.721	.724	.686	.685	.694	.697	.369	.385	.389	.395
arctic-v2	.739	.743	.735	.734	.738	.739	.744	.746	.400	.399	.400	.395

low-relevance coordinates consistently enhances nDCG@10 across varied datasets and encoders, including larger open-source models and challenging zero-shot tasks.

We further offered theoretical insights by framing DIME as a denoising mechanism, where dimension filtering reduces embedding noise while preserving salient semantic signals. This interpretation clarifies why dimension pruning boosts retrieval in practice and guided our development of extensions such as attention-inspired PRF strategies and dynamic retention policies. Notably, DIME operates without any model re-training, making it a lightweight yet powerful addition to existing pipelines.

Future work may explore more adaptive mechanisms for selecting pruning rates per query, integrate multi-turn feedback, or apply DIME to specialized retrieval tasks (e.g., cross-modal search). We believe our findings and open-source artifacts encourage broader adoption of dimension importance estimation, highlighting its potential to improve both retrieval accuracy and efficiency in next-generation semantic search systems.

Acknowledgments

This work was partially supported by projects FAIR (PE0000013) and SERICS (PE00000014) under the MUR National Recovery and Resilience Plan funded by the European Union–NextGenerationEU. Supported also by the project NEREO (Neural Reasoning over Open Data) project funded by the Italian Ministry of Education and Research (PRIN) Grant no. 2022AEF- HAZ.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774* (2023).

[2] Jaime Arguello, Fernando Diaz, Jimmy Lin, and Andrew Trotman. 2015. SIGIR 2015 workshop on reproducibility, inexplicability, and generalizability of results (RIGOR). In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval.* 1147–1148.

[3] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609* (2023).

[4] Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, et al. 2016. Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268* (2016).

[5] Soyuj Basnet, Jerry Gou, Antonio Mallia, and Torsten Suel. 2024. DeeperImpact: Optimizing Sparse Learned Index Structures. *arXiv preprint arXiv:2405.17093* (2024).

[6] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.

[7] Giulio D'Erasmo, Giovanni Trappolini, Nicola Tonellotto, and Fabrizio Silvestri. 2024. ECLIPSE: Contrastive Dimension Importance Estimation with Pseudo-Irrelevance Feedback for Dense Retrieval. *arXiv preprint arXiv:2412.14967* (2024).

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers).* 4171–4186.

[9] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783* (2024).

[10] 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.* 1318–1328.

[11] Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. 2021. SPLADE: Sparse lexical and expansion model for first stage ranking. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval.* 2288–2292.

[12] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948* (2025).

[13] 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.* 113–122.

[14] Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118* (2021).

[15] Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. 2024. Openai o1 system card. *arXiv preprint arXiv:2412.16720* (2024).

[16] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361* (2020).

[17] Aditya Kusupati, Gantavya Bhatt, Aniket Rege, Matthew Wallingford, Aditya Sinha, Vivek Ramanujan, William Howard-Snyder, Kaifeng Chen, Sham Kakade, Prateek Jain, et al. 2022. Matryoshka representation learning. *Advances in Neural Information Processing Systems* 35 (2022), 30233–30249.

[18] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient Memory Management for Large Language Model Serving with PagedAttention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles.*

[19] Jimmy Lin and Xueguang Ma. 2021. A few brief notes on deepimpact, coil, and a conceptual framework for information retrieval techniques. *arXiv preprint arXiv:2106.14807* (2021).

[20] Yi Luan, Jacob Eisenstein, Kristina Toutanova, and Michael Collins. 2021. Sparse, dense, and attentional representations for text retrieval. *Transactions of the Association for Computational Linguistics* 9 (2021), 329–345.

[21] Xueguang Ma, Hengxin Fun, Xusen Yin, Antonio Mallia, and Jimmy Lin. 2023. Enhancing Sparse Retrieval via Unsupervised Learning. In *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region.* 150–157.

[22] Sean MacAvaney, Franco Maria Nardini, Raffaele Perego, Nicola Tonellotto, Nazli Goharian, and Ophir Frieder. 2020. Expansion via prediction of importance with contextualization. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval.* 1573–1576.

[23] Antonio Mallia, Omar Khattab, Torsten Suel, and Nicola Tonellotto. 2021. Learning passage impacts for inverted indexes. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval.* 1723–1727.

[24] Multi-Linguality Multi-Functionality Multi-Granularity. 2024. M3-Embedding: Multi-Linguality, Multi-Functionality, Multi-Granularity Text Embeddings Through Self-Knowledge Distillation. (2024).

[25] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. Ms marco: A human-generated machine reading comprehension dataset. (2016).

[26] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training. (2018).

[27] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners. (2019).

[28] Jack W Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. 2021. Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446* (2021).

[29] Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084* (2019).

[30] Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends® in Information Retrieval* 3, 4 (2009), 333–389.

[31] JJ Rocchio. 1971. Relevance feedback in information retrieval. *The SMART Retrieval System-Experiments in Automatic Document Processing* (1971).

[32] 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 preprint arXiv:2104.08663* (2021).

[33] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971* (2023).

[34] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288* (2023).

[35] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[36] Ellen Voorhees. 2004. Overview of the TREC 2004 Robust Retrieval Track. In *TREC.*

[37] Ellen M. Voorhees. 1996. NIST TREC Disks 4 and 5: Retrieval Test Collections Document Set. doi:10.18434/t47g6m

[38] Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2023. Improving text embeddings with large language models. *arXiv preprint arXiv:2401.00368* (2023).

[39] Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. Multilingual e5 text embeddings: A technical report. *arXiv preprint arXiv:2402.05672* (2024).

[40] Patrick Wilson. 1998. Hjørland, Birger. Information Seeking and Subject Representation: An Activity–Theoretical Approach to Information Science. Westport, Conn.: Greenwood Pr. (New Directions in Information Management, no. 34), 1997. 213p. alk. paper, $59.95 (ISBN 0-313-29893-9). LC 96-51136. *College & Research Libraries* 59, 3 (1998), 287–288. doi:10.5860/crl.59.3.287

[41] Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. 2024. C-pack: Packed resources for general chinese embeddings. In *Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval.* 641–649.

[42] 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 preprint arXiv:2007.00808* (2020).

[43] An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, , et al. 2024. Qwen2 Technical Report. *arXiv preprint arXiv:2407.10671* (2024).

[44] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. 2024. Qwen2.5 Technical Report. *arXiv preprint arXiv:2412.15115* (2024).

[45] Puxuan Yu, Antonio Mallia, and Matthias Petri. 2024. Improved Learned Sparse Retrieval with Corpus-Specific Vocabularies. In *European Conference on Information Retrieval.* Springer, 181–194.

[46] Puxuan Yu, Luke Merrick, Gaurav Nuti, and Daniel Campos. 2024. Arctic-Embed 2.0: Multilingual Retrieval Without Compromise. *arXiv preprint arXiv:2412.04506* (2024).