Dimensionality-Reduction Techniques for Approximate Nearest Neighbor Search: A Survey and Evaluation

Zeyu Wang*, Haoran Xiong, Qitong Wang, Zhenying He, Peng Wang,
Themis Palpanas§, Wei Wang
Shanghai Key Laboratory of Data Science, School of Computer Science, Fudan University
{wangzeyu17, hrxiong20, zhenying, pengwang5, weiwang1}@fudan.edu.cn
†Harvard University, qitong@seas.harvard.edu
§Université Paris Cité, themis@mi.parisdescartes.fr

Abstract

Approximate Nearest Neighbor Search (ANNS) on high-dimensional vectors has become a fundamental and essential component in various machine learning tasks. Recently, with the rapid development of deep learning models and the applications of Large Language Models (LLMs), the dimensionality of the vectors keeps growing in order to accommodate a richer semantic representation. This poses a major challenge to the ANNS solutions since distance calculation cost in ANNS grows linearly with the dimensionality of vectors. To overcome this challenge, dimensionality-reduction techniques can be leveraged to accelerate the distance calculation in the search process. In this paper, we investigate six dimensionality-reduction techniques that have the potential to improve ANNS solutions, including classical algorithms such as PCA and vector quantization, as well as algorithms based on deep learning approaches. We further describe two frameworks to apply these techniques in the ANNS workflow, and theoretically analyze the time and space costs, as well as the beneficial threshold for the pruning ratio of these techniques. The surveyed techniques are evaluated on six public datasets. The analysis of the results reveals the characteristics of the different families of techniques and provides insights into the promising future research directions.

1 Introduction

ANN Search Approximate Nearest Neighbor Search (ANNS) is a crucial component for numerous applications in various fields [13], such as image recognition [22], pose estimation [34], and recommendation systems [11], particularly in high-dimensional spaces. Recent studies have shown that deep neural networks, including large language models, can be augmented by retrieval to enhance accuracy [29, 30] and decrease the magnitude of parameters [21], further emphasizing the significance of ANNS in modern AI applications. Objects, such as images, documents, and videos, can be transformed into dense vectors in the embedding space. ANNS aims to find top-k most similar objects in the embedding space $\mathbb{R}^D$, given a query vector $q \in \mathbb{R}^D$. Compared to the prohibitively high cost of exact search, ANNS is more appealing due to its ability to retrieve high-quality approximate neighbors with a faster response time [14, 15]. To support efficient ANNS, vector indexes are proposed as special data structures to capture the similarity relation between vectors before querying. With vector indexes, most of the data that are irrelevant to the query can be quickly pruned when querying, leading to search efficiency.

Co-first authors
This work was done while Qitong Wang was with Université Paris Cité.
Corresponding author

During ANNS, the time for distance calculation is a major bottleneck. Taking the popular HNSW [33] vector index as an example, the distance calculations account for 60%∼90% of the total query processing time. This is also the case for other types of indexes. The time complexity of common distance metrics, like L2 and inner product, is $\mathcal{O}(D)$ where D is the dimensionality of the vectors. This indicates that the dimensionality of the vectors is a key factor for the efficiency of ANNS vector indexes. In public datasets, the vector dimensionality ranges from hundreds to thousands. With the rapid evolution of pre-trained language models, the dimensionality of embedding vectors also grows: dimensionalities of a few thousands are now commonly used in order to better capture the data semantics [4, 12, 24]. Some of the latest embedding models, such as Alibaba's Qwen2 [51] and Saleforce's SFR [37], produce 3584- and 4096-dimensional vectors, respectively. This presents a significant challenge for the ANNS algorithms.

Dimensionality-Reduction To overcome this challenge, dimensionality-reduction techniques can be leveraged to reduce the distance calculation cost. Specifically, the original high-dimensional vectors can be summarized using lower-dimensional representations, and the distance between these representations can be used to approximate the actual vector distances. The idea behind this is that the accuracy loss when computing the distance will not necessarily decrease the final search accuracy of ANNS. In fact, in the top-k nearest neighbor search problem, only the first k vectors require exact distance, while for the other vectors, we only need to confirm they are farther from the query than the top-k. That is, an estimated distance is already sufficient for most distance calculations in the ANNS problem. Given that the distance estimation achieved by low-dimension representations is usually much more efficient than the full calculation, the performance of current ANNS solutions can be significantly improved. Moreover, since distance calculation is a basic operation for all ANNS algorithms, this approach can benefit all existing algorithms and is orthogonal to specific index structures.

Nevertheless, the estimated distance provided by current dimensionality-reduction techniques cannot be used to safely prune the non-kNN vectors. That is, the distance of some kNN vectors might be over-estimated (and these vectors be skipped), leading to a degraded search accuracy. Recent methods, like ADSampling [18], leverage random projection as the dimensionality-reduction technique to reduce this estimation error, but with a higher estimation cost. On the other hand, many dimensionality-reduction techniques are yet unexplored for the ANNS problem. These techniques, such as Principle Component Analyses (PCA) [2] and Discrete Wavelet Transformation (DWT) [9], are widely studied in the community with solid theoretical foundations, and thus, have the potential to play a positive role in the ANNS problem.

Contributions In this paper, we survey six dimensionality-reduction techniques and two frameworks that apply these techniques to the ANNS problem. The dimensionality-reduction techniques include classical techniques like PCA, machine learning techniques like Product Quantization (PQ) [26], and the deep neural network SEANet [44]. These techniques show advantages in different scenarios in our experiments. We analyze these techniques in theory when serving the ANNS problem. Besides, we study two frameworks to apply these techniques to the ANNS problem, named *in-place transformation* and *out-of-place acceleration*. The former requires pre-processing for the dataset before building the index and adopts an adaptive query algorithm to reduce the cost of distance calculations. The latter constructs an auxiliary data structure when building the index and leverages it to accelerate distance calculations. Furthermore, we implement a pluggable library, named Fudist, to incorporate the dimensionality-reduction techniques as an efficient distance calculator. With Fudist, we evaluate the techniques on 16 real million-scale datasets of different distributions. Based on these empirical results, we list open problems on different technical directions in this research area.

All source codes, datasets, and hyper-parameter settings used in our benchmark are available

online [1]. This ensures the reproducibility of all the experimental results presented in this work. We hope that Fudist will become a standard library for ANNS research that is orthogonal to the index type and search algorithms, and thus, will help improve the comparison of results from different papers.

The rest of this paper is organized as follows. We first review the related works in Section 2. In Section 3, we present the two frameworks for applying dimensionality-reduction techniques to the ANNS problem, and in Section 4, we describe the surveyed dimensionality-reduction techniques. The evaluation results are shown in Section 5, and Section 6 concludes this paper with a list of open problems.

2 Related Work and Background

ANN Indexes State-of-the-art ANN indexes [31] can be categorized into four classes, proximity-graph-based [43, 45], PQ-based[1] [20, 26], Locality Sensitive Hashing (LSH)-based [53, 55] and tree-based [5, 10, 47, 48] (though, some hybrid techniques have started to emerge, combining trees with LSH [50], LSH with proximity-graphs [54], and proximity-graphs with trees [6]; some more recent techniques do not fall in any of these four classes [49]). Among the solutionss in these four classes, graph-based indexes [7] achieve the best query performance for *ng*-approximate (i.e., approximate search with no guarantees [15]) in-memory search, and thus, attracts much interest from the academic and industrial communities. In this paper, we focus on the popular HNSW [33] graph-based index, which is a widely adopted index for ANNS [25, 42], and we implement our dimensionality-reduction techniques on HNSW to verify their effectiveness.

Dimensionality-Reduction Techniques Dimensionality reduction is an important research problem with several solutions proposed in the literature. Classical methods include PCA, DWT, MDS [27], Isomap [40], and MR [36]. These methods leverage linear or nonlinear transformations to obtain low-dimensional representations. We select PCA and DWT as representatives since they can be trained efficiently. Random projection, designed based on Johnson-Lindenstrauss lemma [28], is also widely adopted for dimensionality reduction. In practice, random projection is usually implemented as the inner products of a vector and a group of random vectors, i.e., Locality Sensitive Hashing (LSH). In this paper, we select PM-LSH [55] and ADSampling [18] as representatives of this class.

Some machine learning methods can train a codebook to encode vectors as short codewords and then estimate the actual distance with an efficient asymmetric distance calculation [26]. Represented by Product Quantization (PQ) [26], these methods first segment the vector to obtain several subspaces and then cluster sub-vectors on these subspaces. We select OPQ [20] as an optimized version of PQ for evaluation.

Deep neural networks can also train low-dimensional vectors with the loss of distance deviation. Since no existing models are trained for reducing dimensions in the ANNS problem to the best of our knowledge, we adapt SEANet [44] in ourexperiments to show the potential of this class of methods.

Some other techniques are proposed to reduce vector dimensions for data visualization, like *t*-SNE [41], LargeVis [39], and h-NNE [38]. A recent survey [16] summarizes and evaluates recent progress for these techniques. However, data visualization usually requires a two- or three-dimensional representation, leading to a significant information loss, which makes these techniques impractical for the ANNS problem. Space-filling curves, like the Z-order curve and Hilbert-order curve, can also reduce dimensionality by ordering vectors. Yet, they suffer from the same problem as visualization methods. Dimensionality-reduction techniques also serve for the training and inference of deep neural networks to reduce memory consumption [32]. In this case, the target of the reduced representation is to reserve the model accuracy instead of the distance loss [52], which is out of the scope of this paper.

[1] *PQ* stands for *Product Quantization*.

Algorithm 1: Greedy search (graph G, query q, entry point ep, parameter ef)

1: pq = a priority queue with unlimited capacity, initialized with ep
2: H = a max-heap with capacity ef
3: **while** pq is not empty **do**
4: d_{v_c}, v_c = pop an element from pq
5: $d_{v_{top}}, v_{top}$ = the heap top of H
6: **if** $d_{v_c} > d_{v_{top}}$ **then**
7: break
8: **end if**
9: **for** each neighbor v of v_c which has not been accessed **do**
10: $d_v = dist(v, q)$
11: **if** $d_v < d_{v_{top}}$ **then**
12: Insert (d_v, v) into pq and H
13: **end if**
14: mark v as accessed
15: **end for**
16: resize H to be ef
17: **end while**
18: **return** k smallest elements in H

2.1 Background on Graph-Based Indexes

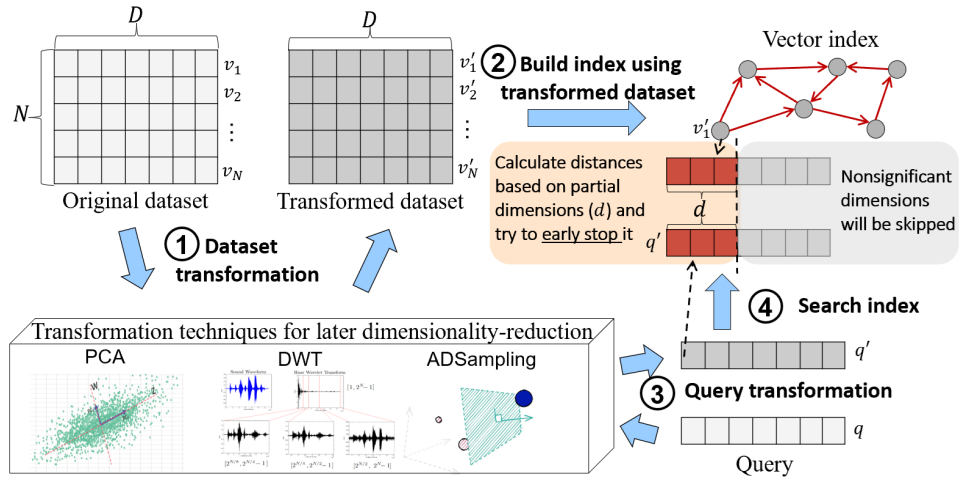
We now describe the greedy search algorithm for graph-based indexes, which we will use later on.

Graph indexes use a directed graph $G(V, E)$ to index vectors, where each vector is represented as a vertex in V, and the edges in E connect vectors based on some kind of proximity. Graph indexes commonly use greedy search to retrieve kNN. As shown in Algorithm 1, the search starts from an entry point ep, which is often selected randomly, and then computes the distance between the neighbors of ep to the query q. The accessed points are stored in a priority queue pq. In the next step, the algorithm selects the closest point to q from pq as the next stop to visit and repeats the above process. Note that not all accessed points can enter pq: the algorithm maintains a size-bounded heap H for best-so-far answers, and only the points that are closer than some point in H are qualified to enter pq. Finally, the algorithm terminates when all the points in pq are farther than the points in H to q. The algorithm is greedy because only points that are relatively close to the query can be accessed. A way to escape such "local optimuma" is to increase the capacity of H, i.e., ef, which is the knob to tune the efficiency-accuracy trade-off in graph indexes.

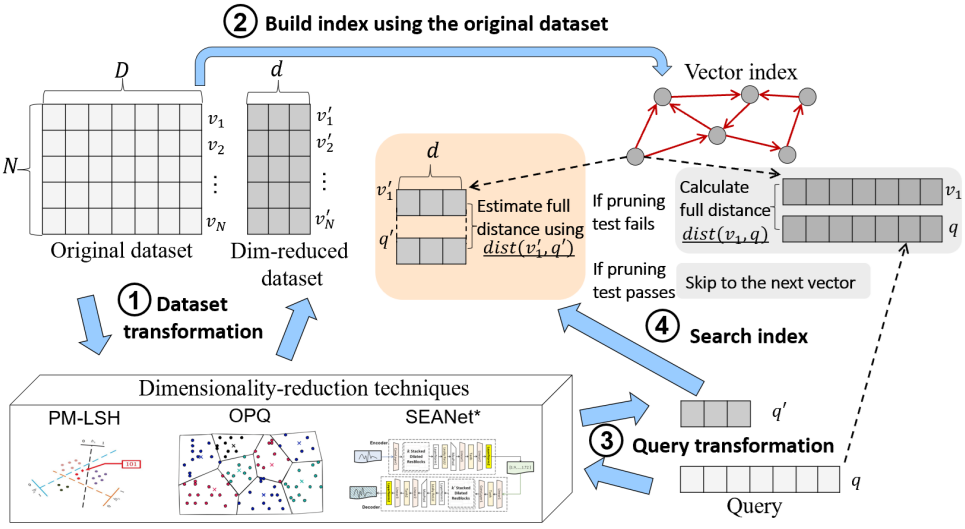
In this paper, we focus on optimizing the seemingly simple distance calculation step (line 10 in Algorithm 1), which is the bottleneck of the query algorithm, as discussed in Section 1.

3 Two Frameworks for Making Use of Dimensionality-Reduction

In this section, we introduce two frameworks to leverage different dimensionality-reduction techniques to benefit the search process: *in-place transformation* described in Section 3.1, and *out-of-place acceleration* described in Section 3.2.



(a) In-place transformation



(b) Out-of-place acceleration

Figure 1: Illustration of two frameworks to apply dimensionality-reduction techniques

3.1 In-Place Transformation

The in-place transformation framework was introduced in [18], for a specific dimensionality-reduction technique, ADSampling. We generalize the framework for all distance-preserved dimensionality-reduction techniques, as shown in Figure 1a. In this sense, any technique that preserves the distances between vectors after the transformation, can be applied to this framework to benefit the ANNS problem. For example, all linear transformations, including PCA, are applicable.

In an in-place transformation framework, the dataset and the query will be transformed in a pre-processing step, before indexing and querying. Then, the vector index will be built based on the transformed dataset. Note that since vector indexes are constructed based on the distances between vectors rather than the vectors themselves, the vector index built from the transformed dataset will be the same as the original one. When querying, the query will be first transformed in the same way. When calculating the distance, the in-place transformation framework adopts an early termination strategy. That is, we calculate the distance dimension by dimension in a cumulative way. Since partial

distances are smaller than full distance, once the current partial distance has exceeded the threshold, the calculation stops, and the rest of the dimensions are skipped. The number of computed dimensions will vary for different vectors (see the incarnadine part in Figure 1a). In the worst case, all the dimensions will be computed and this cost will equal the normal distance calculation cost (besides the cost of termination check).

The rationale behind this method is that transformation techniques will move significant dimensions into the front part of the transformed vector, leading to a higher probabilistic to early terminate the calculation. To reduce the cost of the termination check, a parameter $\Delta > 1$ is introduced as the cycle to check termination. Δ is usually set to 16 or 32 to align with the SIMD instruction width of the distance calculation.

3.2 Out-of-Place Acceleration

The out-of-place acceleration framework leverages an auxiliary data structure, besides the original index structure, to help skip the full distance calculation. As shown in Figure 1b, the vector index is built directly from the raw dataset, and at the same time, a transformed low-dimensional representation table is generated by some dimensionality-reduction technique. This table is the auxiliary data structure. When querying, we generate a low-dimensional representation for the query. For calculating the distance, we first estimate the distance with the low-level representations: if it satisfies the pruning condition, the corresponding full distance will be skipped. Otherwise, the exact distance should be fully calculated.

In the out-of-place acceleration framework, the dimensionality-reduction technique is not required to preserve exactly the original distances between vectors, or thelower bounds of the original distances. This means that some vectors might be skipped incorrectly. Nevertheless, as discussed in Section 1, only the top-k vectors are required to compute the exact distance, while for the other vectors, dismissal or an approximate distance does not necessarily influence the final search accuracy. For example, on graph indexes, there could be several paths from the entry point to the destination kNN. A false dismissal might block one path while other paths are still available. Considering that estimating the distance with low-dimensional representations is usually very efficient, the estimation loss can be compensated by a large parameter ef, which may lead to an overall high search performance. We also introduce a parameter α as a multiplier of the estimated distance to control the influence of the estimation loss.

4 Dimensionality Reduction

In this section, we survey six representative dimensionality reduction techniques from various domains that have the potential to benefit the ANNS problem. The studied techniques are listed in Table 1, including three distance-preserved techniques for the in-place transformation framework (i.e., PCA, DWT, and ADSampling), and three other dimensionality-reduction techniques for the out-of-place acceleration framework (i.e., PM-LSH, SEANet*, and OPQ). In the same table, we also present the additional indexing and query time complexity, as well as the additional space cost.

4.1 Dimensionlity-Reduction Techniques

PCA. Principal Component Analysis (PCA) is a classical linear transformation that selects a new group of vector bases for the data in high-dimensional vector space. Specifically, the eigenvectors of the dataset are selected as the new basis, and the dimensions with larger eigenvalues (or higher variances) are placed at the front positions of the new vector coordinates. Therefore, by evaluating the first few dimensions, we expect to obtain most of the distance, and have a high probability of early stopping the full distance calculation. When applying PCA to the ANNS problem, a transformation matrix is required to reside

Table 1: List of dimensionality-reduction techniques with time and space complexity. N_c is the number of visited points when querying, d is the (expected) dimension of the lower-dimensional representation, Δ is the termination check cycle, X is the number of neurons in SEANet*, m and K_s are the number of segments and codewords in each codebook of OPQ.

Category	Method	Accuracy guarantee	Extra query cost	Extra indexing cost[1]	Extra memory cost
In-place	PCA [2]	exact	$\mathcal{O}(D^2 + N_c d/\Delta)$	$\mathcal{O}(ND^2)$	D^2
	DWT [9]	exact	$\mathcal{O}(D + N_c d/\Delta)$	$\mathcal{O}(ND)$	0
	ADSampling [18]	probabilistic	$\mathcal{O}(D^2 + N_c d/\Delta)$	$\mathcal{O}(ND^2)$	D^2
Out-of-place	PM-LSH [55]	probabilistic	$\mathcal{O}(Dd + N_c d)$	$\mathcal{O}(NDd)$	$Nd + Dd$
	SEANet* [44]	None	$\mathcal{O}(X + N_c d)$	$\mathcal{O}(nX)$	$\mathcal{O}(X)$
	OPQ [20]	None	$\mathcal{O}(DK_s + N_c m)$	$\mathcal{O}(NDK_s)$	$Nm + DK_s + D^2$

[1] The training time is omitted since the size of the training set is smaller than the dataset.

in memory for pre-processing queries, leading to D^2 floats memory cost and $\mathcal{O}(D^2)$ extra query cost. When calculating distances, we check the termination condition for every Δ dimension, and thus the extra distance calculation cost is $\mathcal{O}(d/\Delta)$.

Note that PCA can also be leveraged as a dimensionality-reduction technique in the out-of-place acceleration framework. In this case, we can only take the front part of the transformed dataset and estimate the overall distance with a fixed dimensionality d. However, according to our experiments, using PCA in the out-of-place framework shows inferior performance to that of the in-place transformation framework. In the rest of this paper, when discussing PCA, we refer to using it in the in-place transformation framework.

DWT. Discrete Wavelet Transform (DWT) is a classical time series analysis tool following Parseval's Theorem [9]. It decomposes the vector using hierarchical wavelet transformations, where the major waves are placed in the first positions. The distances between vectors are preserved in both the time and frequency domains, and thus, DWT can be applied in the in-place transformation framework. Compared to PCA, DWT is not a linear transformation and does not need a matrix to be stored in memory, avoiding the extra memory cost. At the same time, it offers linear indexing and querying time costs. Consequently, DWT is the most lightweight technique among the surveyed techniques.

ADSampling. ADSampling adopts a random squared matrix as the transformation matrix, where each element is sampled from a standard Gaussian distribution. As indicated in Johnson-Lindenstrauss lemma [28], the random projection has a shape probabilistic error bound for the deviation of the distance. In this case, the exact distance can be estimated by partial distances with some (probabilistic) confidence. The more dimensions are calculated, the higher the confidence is. Therefore, ADSampling has a probabilistic accuracy guarantee instead of a deterministic one (like PCA and DWT). ADSampling shares the same time and space complexities as PCA, but is much easier to implement, with no training process like SVD in PCA.

PM-LSH PM-LSH is also designed based on the Johnson-Lindenstrauss lemma [28]. In contrast to ADSampling, PM-LSH directly reduces the dimensionality using the inner product between the vector and a group of random vectors (i.e., the hash function family). PM-LSH can encode vectors with ultra-low dimensional representations (e.g. 16) and provides an accuracy guarantee for the distance deviation. Similar to ADSampling, PM-LSH requires a random transformation matrix in memory when querying, leading to Dd space cost and $\mathcal{O}(Dd)$ query pre-processing cost. In addition, in the out-of-place acceleration framework, the dimensionality-reduced dataset (i.e., the data after hashing) requires Nd

space. To generate this auxiliary structure, we additionally need a hashing operation for the dataset when building the index, with $\mathcal{O}(NDd)$ time cost. When calculating a distance, estimating the distance with the low-dimensional representations is necessary, for an extra $\mathcal{O}(d)$ cost. If the pruning condition is not satisfied, the full distance will be calculated.

SEANet*. Series Approximation Network (SEANet) is a deep neural network proposed to represent high-frequency data series with deep learning embeddings. These embeddings are further indexed by the iSAX tree index family [8, 10, 35, 47, 48]. SEANet adopts an encoder-decoder framework with a loss function comprising both distance deviation and vector reconstruction error. We adapt SEANet to the ANNS problem, resulting in SEANet*. Specifically, we remove the decoder part to make SEANet* an encoder-only framework, since vector reconstruction is not necessary in the ANNS problem. Moreover, we remove the construction error in the loss function to focus on distance preservation. Training SEANet introduces a significant time cost when building the index, and it also incurs higher space costs than other alternatives, in order to store the network. Nevertheless, as we will show in the experiments, SEANet* displays a strong potential to improve the query performance significantly.

OPQ. Product Quantization (PQ) is a popular vector compression technique that is commonly adopted along with the IVF index (i.e., iVF-PQ [25]). In the context of a graph-based index, PQ helps direct the search in memory, in order to reduce the I/O cost of a disk-based index, DiskANN [23]. In this paper, we use it to estimate distances efficiently. During indexing, we train the codebooks of the dataset and obtain the codewords of vectors with the codebooks as the auxiliary data structure. When querying, we first generate a distance look-up table using the query and the codebooks. When calculating the distance, the distance can be efficiently estimated by looking up the distance table. Note that OPQ does not give any accuracy guarantee on the distance estimation. Nevertheless, due to the efficiency of distance estimation, the accuracy loss can often be compensated, resulting in an overall performance improvement.

4.2 Beneficial Threshold

We now study at which point the dimensionality-reduction techniques can improve the query performance of the ANNS algorithm, instead of hurting it. Specifically, we focus on a key factor, the pruning ratio ρ, defined as $\rho = 1 - \frac{N_f}{N_c}$, where N_f is the number of full distance calculations and N_c is the number of visited points.

To make our methods more efficient than the raw HNSW, the following inequation should hold:

$$N_c \cdot (C_e + (1-\rho) \cdot \mathcal{O}(D)) + C_p < N_c' \cdot \mathcal{O}(D), \tag{16}$$

where C_e and C_p are the distance estimation cost and query pre-processing cost, respectively (i.e., the second and the first term of the fourth column in Table 1), and N_c' is the number of visited points for the raw HNSW to achieve the same query accuracy. We can derive the *beneficial threshold* of the pruning ratio ρ:

$$\rho > 1 - \frac{N_c'}{N_c} + \frac{1}{\mathcal{O}(D)}(\frac{C_p}{N_c} + C_e). \tag{17}$$

The second term describes the additional search cost of our methods, compared to the raw HNSW, because of the distance estimation deviation. For distance-preserved methods, this term is equal to 1 in theory. For others, this term is smaller than 1. However, in practice, we observe that even for distance-preserved methods, this term is also a bit smaller than 1, due to the transformation and calculation error of float-point numbers. The third term describes the ratio of the amortized cost of distance estimation to the full calculation for each vector. Obviously, with a more accurate and efficient dimensionality-reduction technique, the beneficial threshold will be lower.

Table 2: Dataset characteristics

Datasets	Dataset size	Query size	D	Hardness
Trevi	100,000	200	4096	5335
H&M	105,100	10,000	2048	3439
GIST	1,000,000	10,000	960	12381
MNIST	69,000	200	784	1010
Imagenet	1,000,000	10,000	150	10240
Deep	1,000,000	10,000	96	9451

5 Experiments

5.1 Experimental Settings

Setup Experiments were conducted on an Intel(R) Xeon(R) CPU E5-2620 v4 @ 2.10GHz CPU 20MiB L3 cache with 128GB 2400MHz main memory, running Ubuntu Linux 16.04 LTS. All codes are implemented in C++ and compiled in g++ 9.4.0 with -O3 optimization. SIMD operations are implemented with AVX instructions. The codes are open-sourced in [1].

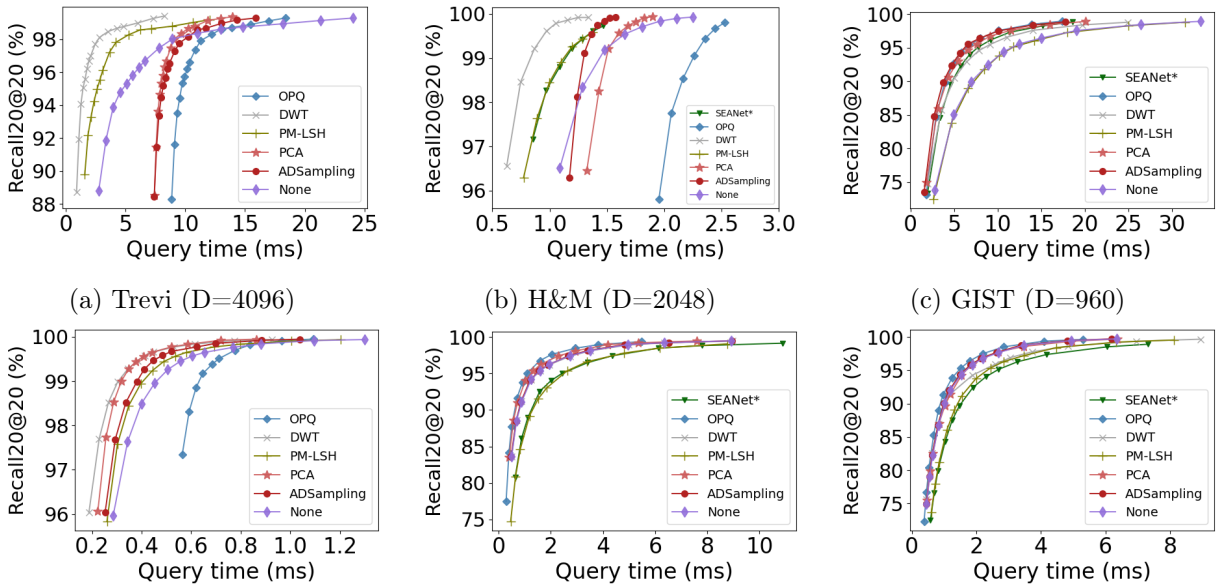
Datasets We select six public datasets of different dimensionality and hardness [46], as listed in Table 2. We generate unbiased workloads consisting of queries with different hardness using the *Steiner*-Hardness method [46] for datasets GIST, H&M, Imagenet, and Deep. We use the public workloads for Trevi and MNIST (their data distributions make them hard to augment with the *Steiner*-Hardness method).

Metrics We use recall to measure the accuracy of ANNS. Formally, $Recall\ k@k = \frac{|A \cap G|}{k}$, where A is the returned approximated neighbors sets and G is the ground truth (i.e., real kNN).

Hyper-parameters k is set to 20, and queries are executed using one thread. The construction parameters of HNSW, M and $efCons$, are tuned to achieve the best query performance. The multiplier α is tuned to make each dimensionality-reduction technique in the out-of-place acceleration framework perform the best. DWT requires the length of the vector to be a power of two, while OPQ requires it to be a multiple of the number of segments. For these two methods, we add padding zeros to the datasets that do not satisfy the corresponding requirements. For DWT, we remove the all-zero columns after thetransformation. For ADSampling, we fix the hyper-parameter ϵ_0 to be 2.1 as recommended, which usually reaches the optimum in our experiments.

5.2 Search Performance

We first disable the utilization of SIMD instructions for distance calculations as in [18], and test the overall search performance of the dimensionality-reduction techniques. The results are shown in Figure 2. For these six datasets, the best alternatives improve the performance of raw HNSW by 6.3x, 2.1x, 2.0x, 1.6x, 1.1x, and 1.1x, respectively under 98% recall, where the best methods are respectively DWT, ADSampling, DWT, and OPQ for the rest three. The performance improvement significantly increases when the dimensionality of the dataset grows to over 700, where the benefit of skipping one-time full-distance calculation is larger. The result of SEANet* on Trevi and MNIST is omitted since the model does not coverage on very high-dimensional (Trevi) and sparse (MNIST) datasets.



(a) Trevi (D=4096) (b) H&M (D=2048) (c) GIST (D=960)

(d) MNIST (D=784) (e) Imagenet (D=150) (f) Deep (D=96)

Figure 2: Query performance with dimensionality-reduction techniques.

On the very high-dimensional dataset Trevi, DWT, and PM-LSH provide significant improvement thanks to their fast preprocessing time, with a complexity of $\mathcal{O}(D)$ and $\mathcal{O}(Dd)$, respectively. On a simple, but with high-dimensional dataset, preprocessing plays an important role in the overall performance. Other methods can only provide an improvement on the high-recall range (>98% recall), where more distance calculations can amortize the preprocessing cost.

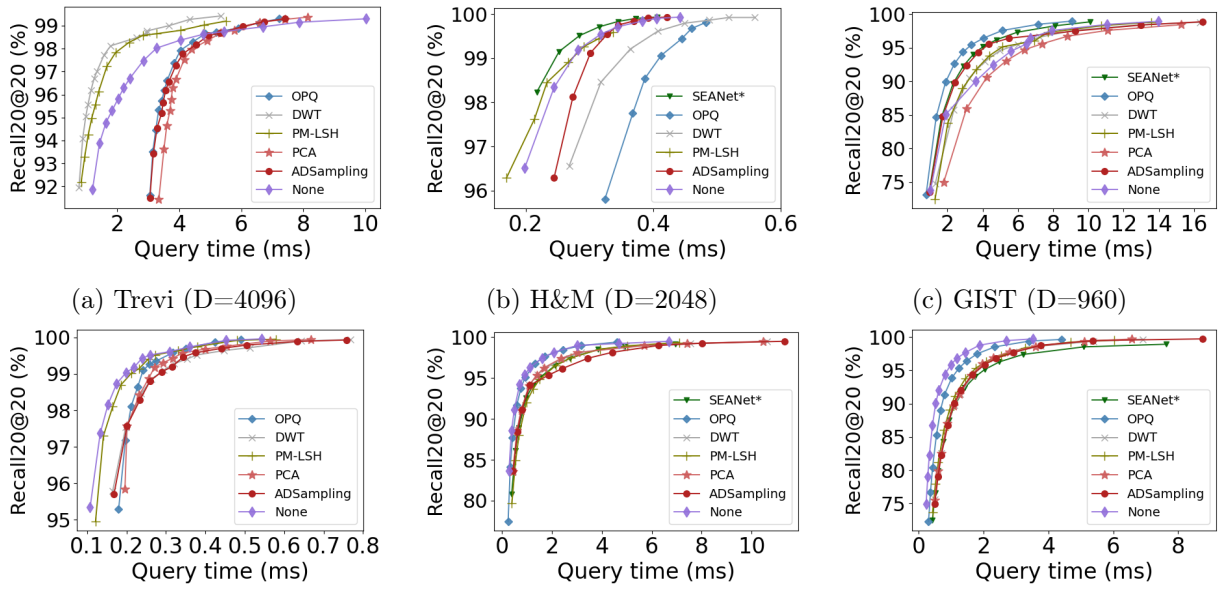
A similar behavior is observed in the H&M dataset in Figure 3b, where DWT and PM-LSH perform the best again. SEANet* achieves outstanding performance that is close to PM-LSH, as well. since the data distribution of this dataset can be well described by the neural network model. OPQ cannot improve the original HNSW performance on this dataset due to its long pre-processing time.

On the GIST dataset, ADSampling, SEANet*, PCA, OPQ, and DWT all show about 1.5x performance improvement over the raw HNSW. Only PM-LSH loses its edge on this dataset. The GIST dataset is a good representative of modern vector embedding datasets, which are dense and have high dimensionality with medium hardness.

MNIST is a sparse dataset where most of the values in the vectors are zeros. In this case, techniques that can effectively summarize the key information like DWT and PCA, show obvious advantages, while OPQ does not perform well since clustering on this sparse dataset tends to be ineffective. Random projections like ADSampling and PM-LSH are generally inferior to DWT and PCA.

On the rest three low-dimensional datasets, OPQ, PCA, and ADSampling show no more than 20% improvements over HNSW, while the other three methods show marginally positive (see Figure 2(d) and (e)) or even negative (see Figure 2(f)) influence on HNSW.

Moreover, we observe that not all techniques outperform the raw HNSW; this is true for all datasets we used in our experiments. This indicates that on some datasets, the pruning ratio of some dimensionality-reduction techniques cannot reach the beneficial threshold. Since no single method consistently outperforms all others, selecting the best technique for different datasets is a necessary step.



(a) Trevi (D=4096) (b) H&M (D=2048) (c) GIST (D=960)

(d) MNIST (D=784) (e) Imagenet (D=150) (f) Deep (D=96)

Figure 3: ANNS performance with SIMD (AVX)

Search with SIMD instructions. Since SIMD instructions are widely adopted in ANNS libraries, such as FAISS and hnswlib, we also test the performance improvement of the surveyed techniques with SIMD (AVX) instructions. The results are shown in Figure 3. For these six datasets, the average query speedup of the best alternatives is 2.2x, 1.2x, 1.7x, 0.9x, 0.9x, and 0.8x, compared to the raw HNSW, for 98% recall. The respective best method is DWT, OPQ, PM-LSH, and OPQ for the rest three.

On the Trevi dataset, all methods lead to an improvement, albeit a small one. OPQ improves more than PCA, since the implementation of the quantization techniques fully leverages the SIMD instructions. On the H&M dataset, only SEANet* improves over the original HNSW across the whole recall range. On the GIST dataset, OPQ isthe best alternative thanks to its high estimation efficiency. SEANet* comes second after OPQ, followed by ADSampling, PM-LSH, and DWT. PCA performs worse than the raw HNSW in this case.

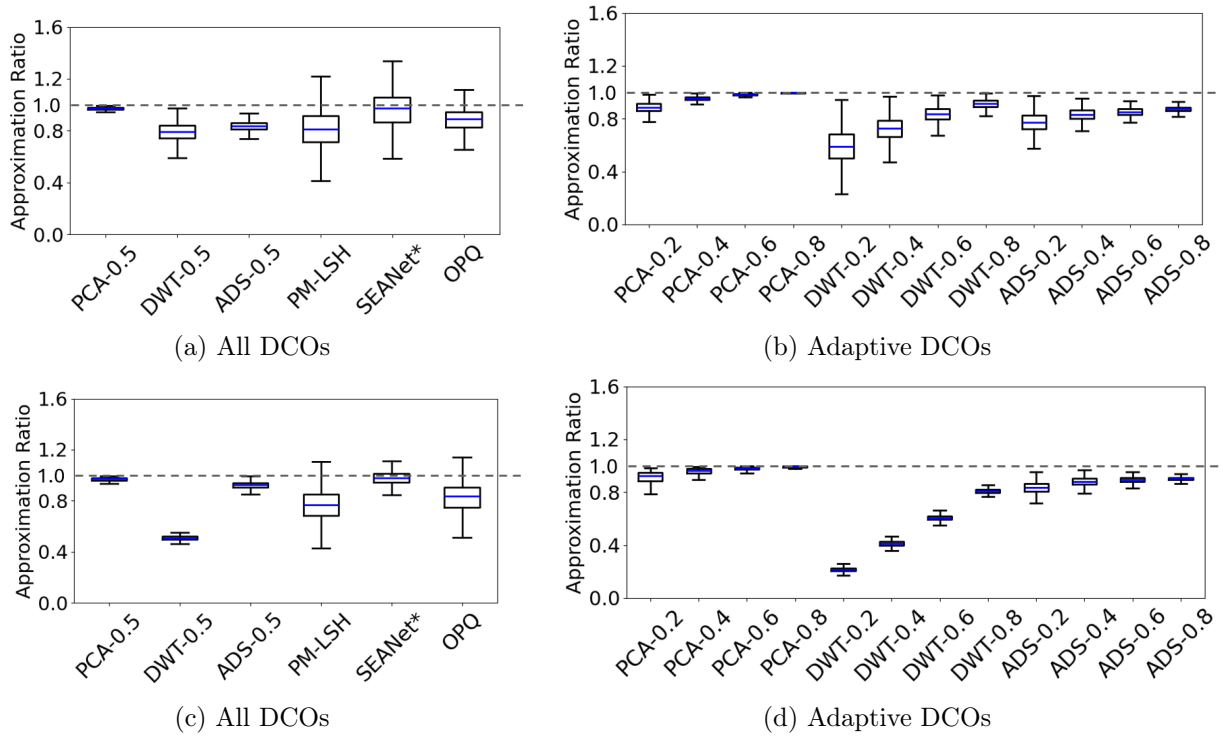
We note that when the dimensionality is smaller than 800 (this includes the MNIST, Imagenet, and Deep datasets), the techniques we evaluated cannot provide a significant improvement over the raw HNSW, since the SIMD acceleration of the raw distance calculations reduces the benefit of the additional pruning that these techniques offer.

5.3 Accuracy Loss

In this section, we study the characteristics of the surveyed dimensionality-reduction techniques by calculating the Approximation Ratio, defined as $AR = \frac{\widehat{dist}(v,q)}{dist(v,q)}$ where $\widehat{dist}$ is the estimated distance. We sample partial vectors from training and test sets to evaluate the approximation ratios.

Figure 4 depicts the distribution of AR on GIST and H&M datasets, where the dotted line indicates $AR = 1.0$, an optimal case of distance-estimation techniques. We use the notation PCA-0.5 to indicate that weestimate distances using the first 50% of the dimensions in the transformed dataset.

As distance-preserved techniques, the AR of PCA and DWT is always below 1; PCA is much tighter than DWT, with a smaller variance. Similarly, although ADSampling does not provide a lower bound, it hardly generates false dismissals thanks to itsreliable probabilistic guarantee. On the H&M dataset



(a) All DCOs

(b) Adaptive DCOs

(c) All DCOs

(d) Adaptive DCOs

Figure 4: Approximation ratio on GIST (top) and H&M (bottom)

(refer to Figure 4c), SEANet* provides much more accurate estimation than PM-LSH and OPQ. This result indicates the strong potential of deep learning methods to estimate the similarity between very high-dimensional vectors by using low-dimensional representations. As shown in Figures 4b and 4d, when applying dimensionality-reduction techniques in the in-place transformation framework, we can expect a more accurate estimation with a smaller variance by checking more dimensions. For PCA, we can already get a very accurate estimation by calculating 80% of the dimensions when querying. Although DWT shows inferior performance than the other techniques, with no more than 50% of the dimensions, it quickly turns accurate with more dimensions (>60%).

5.4 Verification of Beneficial Threshold

In the final experiment, we calculate each term in Equation 17 to verify the effectiveness of the beneficial threshold, and evaluate the practical cost of the asymptotic complexities reported in Table 1. We report the results on the Deep and GIST datasets, in Table 3. On the Deep dataset, it is only for OPQ that the pruning ratio ρ exceeds the beneficial threshold θ, while on the GIST dataset, all the methods meet the threshold. This verifies the experimental results shown in Figure 2. On the Deep dataset, the beneficial thresholds for distance-preserved techniques are very high, and for PCA it is larger than 1, which indicates it is impossible to gain performance improvement. On the GIST dataset, the increase of the practical cost of $\mathcal{O}(D)$ leads to a major drop of the beneficial threshold θ. Moreover, according to the value of $1 - \frac{N_c'}{N_c}$, we observe that the extra search cost introduced by the accuracy loss occupies a small portion of θ. Except for the very high-dimensional and simple dataset Trevi, the amortized preprocessing cost, $\frac{C_p}{N_c \mathcal{O}(D)}$ is also very small. In this case, the estimation cost $\frac{C_e}{\mathcal{O}(D)}$ is the most important factor. We observe that OPQ wins the first place on both datasets w.r.t. the estimation efficiency. We also note that on the GIST dataset, the N_c value of SEANet* is close to that of the distance-preserved methods,

Table 3: Key metrics evaluation on Deep and GIST on recall = 0.94. N_c' is the number of visited points for the raw HNSW, and $\mathcal{O}(D)$ is the cost of full distance calculation. N_c is the number of visited points for corresponding dimensionality-reduction techniques, C_e and C_p are the distance estimation cost and query pre-processing cost, ρ and θ are the pruning ratio and the beneficial threshold.

Recall@0.94	**Deep**. $N_c' = 6760.3$, $\mathcal{O}(D) \approx 0.2423\ \mu s$					**GIST**. $N_c' = 7929.5$, $\mathcal{O}(D) \approx 1.305\ \mu s$				
	N_c	$C_e\ (\mu s)$	$C_p\ (\mu s)$	ρ	θ	N_c	$C_e\ (\mu s)$	$C_p\ (\mu s)$	ρ	θ
ADSampling	6760.1	0.2354	2.939	0.9301	**0.9734**	8165.3	0.6327	194.4	**0.9383**	0.5319
DWT	6813.1	0.2348	1.066	0.9423	**0.9775**	7812.2	0.8935	6.836	**0.9399**	0.6703
PCA	6821.7	0.2420	2.956	0.9424	**1.0093**	7773.3	0.7439	195.2	**0.9402**	0.5691
PM-LSH	7185.4	0.1300	0.681	0.1601	**0.5960**	8538.8	0.2200	7.713	**0.2938**	0.2406
SEANet*	8433.9	0.1230	14.48	0.2616	**0.7131**	7836.9	0.1471	16.59	**0.4828**	0.1024
OPQ	7148.8	0.0768	19.22	**0.5703**	0.3832	8274.5	0.1194	371.9	**0.5941**	0.1676

which indicates a strong potential for the estimation effectiveness of deep learning methods.

6 Conclusions and Future Directions

In this paper, we survey six dimensionality-reduction techniques that have the potential to benefit the query performance of the ANNS algorithm. We employ two frameworks, in-place transformation, and out-of-place acceleration, to integrate these techniques into the ANNS indexing and querying workflow. Under these frameworks, we study the theoretical time and space complexity and benchmark their performance with an extensive and fair evaluation. The results indicate that the best alternatives improve the query performance of the original HNSW by up to 6x, but at the same time, the performance of each technique varies widely across different datasets, and in some cases, we observe no improvement at all.

We observe that at the framework level, the out-of-place acceleration framework offers greater flexibility for use with pre-existing graph indexes, as it can enhance query performance through the addition of an auxiliary data structure. In contrast, the in-place transformation framework necessitates the reconstruction of the index, but with lower memory usage compared to the out-of-place framework.

Based on the results of our study, we discuss below promising research directions.

1. Quantization techniques with quality guarantees. Due to the high efficiency of distance estimation, OPQ shows a robust performance improvement in the high-recall range. However, since the distance estimation does not come with any guarantees, it takes much time to tune the parameters to achieve the optimal trade-off between efficiency improvement and accuracy loss. In this case, an accuracy guarantee, which can be provided by LSH and the techniques of the in-place transformation framework, will make PQ more feasible and efficient for the ANNS problem [17, 19].

2. Deep neural networks show a strong potential for efficient low-dimensional representations. Under the same dimensionality, deep learning methods show the best estimation accuracy over all methods on some datasets. There is still a large design space for deep learning methods w.r.t. the model framework, loss function, sampling, and training strategy, in this scenario. Moreover, finding the optimal hyper-parameters, such as the dimensionality of the representations, is a challenging open problem.

3. Adaptive dimensionality-reduction techniques selection. There is no single technique outperforming all the others according to our evaluation. This indicates that an effective method selection approach is necessary in practice. One of the challenges is how to relate the accuracy loss of the estimation with the search effort in the ANNS problem, which relies on the cost analysis of the

graph search algorithm [46].

4. Efficient storage compression with dimensionality-reduction techniques. As the dimensionality of the vectors grow larger, the storage cost of vectors becomes very high, which renders current database storage and query optimization techniques ineffective for the vector type. While scalar quantization techniques [3] provide an alternative, dimensionality-reduction offers an orthogonal approach to address this problem. In this sense, combining these two types of techniques may offer an efficient database storage compression solution for vector data.

Acknowledgments

Supported by EU Horizon projects AI4Europe (101070000), TwinODIS (101160009), ARMADA (101168951), DataGEMS (101188416), RECITALS (101168490).

References

[1] Fudist. `https://github.com/CaucherWang/Fudist`, 2023. Accessed: 2024-10-30.

[2] Hervé Abdi and Lynne J. Williams. Principal component analysis. *WIREs Computational Statistics*, 2(4):433–459, 2010.

[3] Cecilia Aguerrebere, Ishwar Singh Bhati, Mark Hildebrand, Mariano Tepper, and Theodore Willke. Similarity search in the blink of an eye with compressed indices. *Proc. VLDB Endow.*, 16(11):3433–3446, jul 2023.

[4] Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, et al. The falcon series of open language models. *arXiv preprint arXiv:2311.16867*, 2023.

[5] Akhil Arora, Sakshi Sinha, Piyush Kumar, and Arnab Bhattacharya. Hd-index: Pushing the scalability-accuracy boundary for approximate knn search in high-dimensional spaces. *Proc. VLDB Endow.*, 11(8):906–919, apr 2018.

[6] Ilias Azizi, Karima Echihabi, and Themis Palpanas. Elpis: Graph-based similarity search for scalable data science. *Proc. VLDB Endow.*, 16(6):1548–1559, apr 2023.

[7] Ilias Azizi, Karima Echihabi, and Themis Palpanas. Graph-based vector search: An experimental evaluation of the state-of-the-art. *Proceedings of the ACM Management of Data (PACMMOD)*, 2025.

[8] Alessandro Camerra, Themis Palpanas, Jin Shieh, and Eamonn Keogh. isax 2.0: Indexing and mining one billion time series. In *2010 IEEE International Conference on Data Mining*, pages 58–67, 2010.

[9] Kin-Pong Chan and Ada Wai-Chee Fu. Efficient time series matching by wavelets. In *Proceedings 15th International Conference on Data Engineering (Cat. No.99CB36337)*, pages 126–133, 1999.

[10] Manos Chatzakis, Panagiota Fatourou, Eleftherios Kosmas, Themis Palpanas, and Botao Peng. Odyssey: A journey in the land of distributed data series similarity search. *Proceedings of the VLDB Endowment*, 16(5):1140–1153, 2023.

[11] Abhinandan S. Das, Mayur Datar, Ashutosh Garg, and Shyam Rajaram. Google news personalization: Scalable online collaborative filtering. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, page 271–280. Association for Computing Machinery, 2007.

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

[13] Karima Echihabi, Kostas Zoumpatianos, and Themis Palpanas. Scalable machine learning on high-dimensional vectors: From data series to deep network embeddings. In *International Conference on Web Intelligence, Mining and Semantics (WIMS)*, pages 1–6. ACM, 2020.

[14] Karima Echihabi, Kostas Zoumpatianos, Themis Palpanas, and Houda Benbrahim. The lernaean hydra of data series similarity search: An experimental evaluation of the state of the art. *Proc. VLDB Endow.*, 12(2):112–127, 2018.

[15] Karima Echihabi, Kostas Zoumpatianos, Themis Palpanas, and Houda Benbrahim. Return of the lernaean hydra: Experimental evaluation of data series approximate similarity search. *Proc. VLDB Endow.*, 13(3):403–420, 2019.

[16] Mateus Espadoto, Rafael M Martins, Andreas Kerren, Nina ST Hirata, and Alexandru C Telea. Toward a quantitative survey of dimension reduction techniques. *IEEE transactions on visualization and computer graphics*, 27(3):2153–2173, 2019.

[17] Jianyang Gao, Yutong Gou, Yuexuan Xu, Yongyi Yang, Cheng Long, and Raymond Chi-Wing Wong. Practical and asymptotically optimal quantization of high-dimensional vectors in euclidean space for approximate nearest neighbor search, 2024.

[18] Jianyang Gao and Cheng Long. High-dimensional approximate nearest neighbor search: with reliable and efficient distance comparison operations. *Proc. ACM Manag. Data*, 1(1), may 2023.

[19] Jianyang Gao and Cheng Long. Rabitq: Quantizing high-dimensional vectors with a theoretical error bound for approximate nearest neighbor search. *Proc. ACM Manag. Data*, 2(3), may 2024.

[20] Tiezheng Ge, Kaiming He, Qifa Ke, and Jian Sun. Optimized product quantization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(4):744–755, 2014.

[21] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. Retrieval augmented language model pre-training. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 3929–3938. PMLR, 13–18 Jul 2020.

[22] Young Kyun Jang and Nam Ik Cho. Generalized product quantization network for semi-supervised image retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[23] Suhas Jayaram Subramanya, Fnu Devvrit, Harsha Vardhan Simhadri, Ravishankar Krishnawamy, and Rohan Kadekodi. Diskann: Fast accurate billion-point nearest neighbor search on a single node. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[24] Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

[25] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547, 2019.

[26] Herve Jégou, Matthijs Douze, and Cordelia Schmid. Product quantization for nearest neighbor search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(1):117–128, 2011.

[27] Joseph B Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–27, 1964.

[28] Kasper Green Larsen and Jelani Nelson. The johnson-lindenstrauss lemma is optimal for linear dimensionality reduction. *arXiv preprint arXiv:1411.2404*, 2014.

[29] 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. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc., 2020.

[30] Huayang Li, Yixuan Su, Deng Cai, Yan Wang, and Lemao Liu. A survey on retrieval-augmented text generation. *arXiv preprint arXiv:2202.01110*, 2022.

[31] Wen Li, Ying Zhang, Yifang Sun, Wei Wang, Mingjie Li, Wenjie Zhang, and Xuemin Lin. Approximate nearest neighbor search on high dimensional data — experiments, analyses, and improvement. *IEEE Transactions on Knowledge and Data Engineering*, 32(8):1475–1488, 2020.

[32] Fuyuan Lyu, Xing Tang, Hong Zhu, Huifeng Guo, Yingxue Zhang, Ruiming Tang, and Xue Liu. Optembed: Learning optimal embedding table for click-through rate prediction. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 1399–1409, 2022.

[33] Yu A. Malkov and D. A. Yashunin. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(4):824–836, 2020.

[34] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 2161–2168, 2006.

[35] Botao Peng, Panagiota Fatourou, and Themis Palpanas. Fast data series indexing for in-memory data. *VLDB J.*, 30(6):1041–1067, 2021.

[36] Aniket Rege, Aditya Kusupati, Alan Fan, Qingqing Cao, Sham Kakade, Prateek Jain, Ali Farhadi, et al. Adanns: A framework for adaptive semantic search. *Advances in Neural Information Processing Systems*, 36:76311–76335, 2023.

[37] Meng Rui, Liu Ye, Shafiq Rayhan Joty, Xiong Caiming, Zhou Yingbo, and Semih Yavuz. Sfr-embedding-2: Advanced text embedding with multi-stage training, 2024.

[38] Saquib Sarfraz, Marios Koulakis, Constantin Seibold, and Rainer Stiefelhagen. Hierarchical nearest neighbor graph embedding for efficient dimensionality reduction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 336–345, 2022.

[39] Jian Tang, Jingzhou Liu, Ming Zhang, and Qiaozhu Mei. Visualizing large-scale and high-dimensional data. In *Proceedings of the 25th international conference on world wide web*, pages 287–297, 2016.

[40] Joshua B Tenenbaum, Vin de Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500):2319–2323, 2000.

[41] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[42] Jianguo Wang, Xiaomeng Yi, Rentong Guo, Hai Jin, Peng Xu, Shengjun Li, Xiangyu Wang, Xiangzhou Guo, Chengming Li, Xiaohai Xu, Kun Yu, Yuxing Yuan, Yinghao Zou, Jiquan Long, Yudong Cai, Zhenxiang Li, Zhifeng Zhang, Yihua Mo, Jun Gu, Ruiyi Jiang, Yi Wei, and Charles Xie. Milvus: A purpose-built vector data management system. In *Proceedings of the 2021 International Conference on Management of Data*, SIGMOD '21, page 2614–2627, New York, NY, USA, 2021. Association for Computing Machinery.

[43] Mengzhao Wang, Xiaoliang Xu, Qiang Yue, and Yuxiang Wang. A comprehensive survey and experimental comparison of graph-based approximate nearest neighbor search. *Proc. VLDB Endow.*, 14(11):1964–1978, jul 2021.

[44] Qitong Wang and Themis Palpanas. Deep learning embeddings for data series similarity search. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, page 1708–1716. ACM, 2021.

[45] Zeyu Wang, Peng Wang, Themis Palpanas, and Wei Wang. Graph-and tree-based indexes for high-dimensional vector similarity search: Analyses, comparisons, and future directions. *IEEE Data Eng. Bull.*, 46(3):3–21, 2023.

[46] Zeyu Wang, Qitong Wang, Xiaoxing Cheng, Peng Wang, Themis Palpanas, and Wei Wang. *steiner*-hardness: A query hardness measure for graph-based ann indexes. *arXiv preprint arXiv:2408.13899*, 2024.

[47] Zeyu Wang, Qitong Wang, Peng Wang, Themis Palpanas, and Wei Wang. Dumpy: A compact and adaptive index for large data series collections. *Proc. ACM Manag. Data*, 1(1), may 2023.

[48] Zeyu Wang, Qitong Wang, Peng Wang, Themis Palpanas, and Wei Wang. Dumpyos: A data-adaptive multi-ary index for scalable data series similarity search. *The VLDB Journal*, pages 1–25, 2024.

[49] Jiuqi Wei, Xiaodong Lee, Zhenyu Liao, Themis Palpanas, and Botao Peng. Subspace collision: An efficient and accurate framework for high-dimensional approximate nearest neighbor search. *Proceedings of the ACM Management of Data (PACMMOD)*, 2025.

[50] Jiuqi Wei, Botao Peng, Xiaodong Lee, and Themis Palpanas. DET-LSH: A locality-sensitive hashing scheme with dynamic encoding tree for approximate nearest neighbor search. *Proc. VLDB Endow.*, 17(9):2241–2254, 2024.

[51] An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, and Chang Zhou et al. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024.

[52] Hailin Zhang, Penghao Zhao, Xupeng Miao, Yingxia Shao, Zirui Liu, Tong Yang, and Bin Cui. Experimental analysis of large-scale learnable vector storage compression. *Proceedings of the VLDB Endowment*, 17(4):808–822, 2023.

[53] Xi Zhao, Yao Tian, Kai Huang, Bolong Zheng, and Xiaofang Zhou. Efficient index construction and approximate nearest neighbor search in high-dimensional spaces. *Proc. VLDB Endow.*, 16(8):1979–1991, 2023.

[54] Xi Zhao, Yao Tian, Kai Huang, Bolong Zheng, and Xiaofang Zhou. Towards efficient index construction and approximate nearest neighbor search in high-dimensional spaces. *Proc. VLDB Endow.*, 16(8):1979–1991, 2023.

[55] Bolong Zheng, Xi Zhao, Lianggui Weng, Nguyen Quoc Viet Hung, Hang Liu, and Christian S. Jensen. Pm-lsh: A fast and accurate lsh framework for high-dimensional approximate nn search. *Proc. VLDB Endow.*, 13(5):643–655, jan 2020.