Efficient Data-aware Distance Comparison Operations for High-Dimensional Approximate Nearest Neighbor Search

Liwei Deng, Penghao Chen, Ximu Zeng
University of Electronic Science and Technology of China
deng_liwei,penghaochen,ximuzeng@std.uestc.edu.cn

Tianfu Wang
University of Science and Technology of China
tianfuwang@mail.ustc.edu.cn

Yan Zhao
University of Electronic Science and Technology of China✉
yanz@cs.aau.dk

Kai Zheng*
University of Electronic Science and Technology of China✉
zhengkai@uestc.edu.cn

ABSTRACT

High-dimensional approximate K nearest neighbor search (AKNN) is a fundamental task for various applications, including information retrieval. Most existing algorithms for AKNN can be decomposed into two main components, i.e., candidate generation and distance comparison operations (DCOs). While different methods have unique ways of generating candidates, they all share the same DCO process. In this study, we focus on accelerating the process of DCOs that dominates the time cost in most existing AKNN algorithms. To achieve this, we propose an Data-Aware Distance Estimation approach, called *DADE*, which approximates the *exact* distance in a lower-dimensional space. We theoretically prove that the distance estimation in *DADE* is *unbiased* in terms of data distribution. Furthermore, we propose an optimized estimation based on the unbiased distance estimation formulation. In addition, we propose a hypothesis testing approach to adaptively determine the number of dimensions needed to estimate the *exact* distance with sufficient confidence. We integrate *DADE* into widely-used AKNN search algorithms, e.g., *IVF* and *HNSW*, and conduct extensive experiments to demonstrate the superiority.

PVLDB Reference Format:
Liwei Deng, Penghao Chen, Ximu Zeng, Tianfu Wang, Yan Zhao, and Kai Zheng. Efficient Data-aware Distance Comparison Operations for High-Dimensional Approximate Nearest Neighbor Search. PVLDB, 18(3): 812 - 821, 2024.

doi:10.14778/3712221.3712244

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://github.com/Ur-Eine/DADE.

1 INTRODUCTION

Searching for the K nearest neighbors (KNN) in the high-dimensional Euclidean space is pivotal for various applications, such as data mining [13, 16, 17, 48], information retrieval [23, 26], scientific computing, recommendation systems [26, 35, 42], and large language models [2, 8, 34]. However, with the increase of dimensions, the performance of existing indexing algorithms such as R-Tree [5, 25] and KD-Tree [6, 47] for exact searching degrades to that of brute-force search, which is time-consuming and often fails to meet the practical application requirements. This phenomenon is known as the curse of dimensionality. Thus, to achieve the tradeoff between accuracy and latency, existing research mainly focuses on developing algorithms for approximate K nearest neighbors (AKNN) search, which aim to provide neighbors with acceptable recall while significantly improving search speed.

Existing studies for AKNN can be broadly categorized into four main approaches: (1) graph-based [9, 18, 27, 39, 41, 50, 52], (2) quantization-based [4, 21, 22, 29, 31, 36], (3) tree-based [7, 12, 14, 40], and (4) hashing-based [15, 19, 46, 53]. Despite their different methodologies, these algorithms generally follow a common paradigm: they first generate candidate neighbors and then refine these candidates to identify the K nearest neighbors. It should be noted that these algorithms share a similar process in the refinement process, which involves maintaining a max-heap Q to store the current KNNs. For each new candidate, they check whether its distance to the query is less than the maximum in Q [20]. If the distance is not less than the maximum, the candidate is discarded. Otherwise, the max-heap Q is updated to remove the object with the maximum distance and add the new candidate. This checking process is called *distance comparison operation* (DCO) [20]. DCO is widely adopted in AKNN approaches. For example in HNSW and IVF, they first compute the *exact* distance between the query and a candidate, and then compare this distance to the maximum in Q. The computation in the full-dimensional space has a time complexity of $O(D)$, where D is the number of dimensions in the Euclidean space. Gao et al. [20] demonstrate that the cost of performing DCOs dominates the total query time of *HNSW*. For example, on the DEEP dataset with 256 dimensions, DCOs take 77.2% of the total running time.

An intuitive idea to speed up the process of DCOs is to determine the distance between query and candidate without the calculation of the exact distance. This goal can be achieved through some distance approximation methods such as product quantization [22, 29]. However, despite these methods can enhance the efficiency of distance computation, the accuracy will degrade dramatically. Thus, these methods are more suitable for generating candidates rather than obtaining KNN from the generated candidates [49]. Recently, Gao et al. [20] systematically study solutions for DCOs and propose a method called *ADSampling*, which firstly applies a *random*

*Liwei Deng and Penghao Chen are equally contributed to this work.✉ Kai Zheng is with the School of Computer Science and Engineering, University of Electronic Science and Technology of China. Yan Zhao and Kai Zheng are with Shenzhen Institute for Advanced Study, University of Electronic Science and Technology of China.

Proceedings of the VLDB Endowment, Vol. 18, No. 3 ISSN 2150-8097.
doi:10.14778/3712221.3712244

orthogonal transformation [11] to the original vectors and then adaptively determines the number of dimensions to be sampled for each object during the query phase based on the DCO for distance estimation. They provide a theoretical proof to demonstrate that the distance estimation of *ADSampling* is *unbiased* in terms of the random transformation and that the failure probability is bounded by a constant. However, *ADSampling* is data-oblivious and cannot provide an accurate distance estimation for a specific dataset, which hinders it from being optimal.

In this study, we propose a new method called *DADE* for Data-Aware Distance estimation [1]. Specifically, *DADE* utilizes an *orthogonal transformation* to rotate the original space and performs DCOs in the projected space. The number of dimensions d ($d \leq D$) is adaptively determined to estimate the distance between the query and individual candidates. Different from previous distance estimation methods that project all objects with *equal* dimensions, *DADE* is more similar to *ADSampling* in that it employs a different number of dimensions for DCOs on different objects. However, *DADE* improves upon *ADSampling* by deriving the orthogonal transformation based on data distribution rather than *randomly*, leading to more accurate distance estimation within the same running time. We provide theoretical proofs for two key points: (1) our distance estimation is *unbiased* in terms of data distribution regardless of the number of dimensions, and (2) compared to other *unbiased* distance estimation methods, our approach is optimized in terms of variance. These proofs demonstrate that *DADE* is a better distance estimation method when the transformation is *orthogonal*. Furthermore, to determine when to expand the number of dimensions, a hypothesis testing approach is adopted, in which the significance level is controlled by a probability defined and empirically derived from the data objects.

In summary, we conclude our contributions as follows.

- We propose a new method *DADE*, which can be integrated as a plug-in component to accelerate the search process in existing AKNN algorithms such as *IVF* and *HNSW*.
- We provide a theoretical proof showing that the proposed distance estimation is *unbiased* and *optimized* in terms of data distribution when the transformation is *orthogonal*.
- We propose a hypothesis testing method to *adaptively* control the number of dimensions used in distance estimation. The probability of estimation deviation is empirically approximated from the data objects, addressing the challenge of explicitly expressing the data distribution.
- We conduct extensive experiments on real datasets to show the superiority of our method. For example, on the DEEP dataset, *DADE* improves the queries per second (QPS) by over 40% on *HNSW* compared with the state-of-the-art approach *ADSampling*, while maintaining the same level of accuracy.

2 PRELIMINARIES

We proceed to present the necessary preliminaries and then define the problem addressed.

Lemma 1. *Assume $X_1, X_2 \in \mathbb{R}^D$ are independent and identically distributed random vectors, and $Y_i = X_i - \mathbb{E}[X_i]$ $(for\ i = 1, 2)$. For $W_D \in \mathbb{R}^{D\times D}, {W_D}^T W_D = \mathbf{I}$, the following equation holds:*

$$\|{W_D}^T Y_1 - {W_D}^T Y_2\|_2^2 = \|{W_D}^T X_1 - {W_D}^T X_2\|_2^2 \tag{1}$$

Proof. $\|{W_D}^T Y_1 - {W_D}^T Y_2\|_2^2 = \|{W_D}^T (Y_1 - Y_2)\|_2^2$
$= \|{W_D}^T [(X_1 - \mathbb{E}[X_1]) - (X_2 - \mathbb{E}[X_2])]\|_2^2$
$= \|{W_D}^T (X_1 - X_2)\|_2^2 = \|{W_D}^T X_1 - {W_D}^T X_2\|_2^2$. □

Thus, without loss of generalization, we assume that the random vector in the subsequent proof is zero mean, i.e., $\mathbb{E}[X] = 0$. The vectors used in this paper are all column vectors. In addition, we also provide a simple Lemma about *orthogonal projection*, which will be used as a proof part in the following sections.

Lemma 2. *Orthogonal projection does not change the sum of variances of all dimensions.*

Proof. $\sum_{k=1}^{D} \mathrm{Var}(x_k) = \mathbb{E}[X^T X] - \mathbb{E}[X]^T \mathbb{E}[X]$
$= \mathbb{E}[X^T W_D {W_D}^T X] = \sum_{k=1}^{D} \mathrm{Var}({w_k}^T X)$.
Where x_k is the component of the zero-mean random vector X in the k-th dimension. □

Definition 1 (Distance Comparison Operation [20]). *Given a query $\boldsymbol{q}$, an object $\boldsymbol{o}$ and a distance threshold r, the **distance comparison operation** (DCO) is to decide whether object $\boldsymbol{o}$ has its distance dis to $\boldsymbol{q}$ no greater than r and if so, return dis.*

3 METHODOLOGY

We develop a new method called *DADE* to perform DCOs with better efficiency than existing approaches. Specifically, *DADE* first rotates the original Euclidean space with a data-aware orthogonal transformation and then conducts DCOs based on the projected space, in which the distance between a query and a candidate is estimated in a subspace with *fewer* dimensions for better efficiency. Compared with existing studies, *DADE* shows three main differences: (1) Compared with random projection methods [15, 19, 45] that project objects into vectors with *equal* dimensions, which may be knowledge demanding and difficult to set in practice [20], *DADE* estimates the distance with *adaptive* dimensions. (2) Compared with the SOTA method, i.e., *ADSampling* [20], the transformation in *DADE* is data-aware, which provides a more accurate distance approximation with the same level of running time. (3) The number of dimensions to be used for distance computation is determined by hypothesis testing, in which the *unknown* data distribution for the hypothesis testing is empirically approximated. The details of *DADE* are elaborated in the following sections.

3.1 Unbiased Estimation

An intuitive idea to speed up the process of DCOs is to determine whether $dis \leq r$ without computing the exact distance between the query and the candidate. Thus, a set of distance approximation methods can be adopted. For example, *ADSampling* conducts *randomly orthogonal transformation* on an object with a random matrix $W_d \in \mathbb{R}^{D\times d}$ where $d \leq D$, and then perform the DCOs with the approximate distance. However, these methods are data-oblivious, which prevents them from optimum. To deal with this, we first provide a data-aware *unbiased* estimation as the following Lemma.

Lemma 3. *Given a set of orthogonal bases in $\mathbb{R}^D$ Euclidean space, i.e., $W_d := [w_1, w_2, \cdots, w_d] \in \mathbb{R}^{D\times d}$, where $\forall i \neq j$, ${w_i}^T w_j = 0$ and*

[1] By distances, we refer to the Euclidean distance without further specification.

${w_i}^T w_i = 1$. *Assume* X, X_1, *and* $X_2 \in \mathbb{R}^D$ *are three independent and identically distributed random vectors. The following equation holds:*

$$\mathbb{E}[\|X_1 - X_2\|_2^2] = \frac{\sigma^2(1,D)}{\sigma^2(1,d)}\mathbb{E}[\|{W_d}^T X_1 - {W_d}^T X_2\|_2^2] \tag{2}$$

where $\sigma^2(i,j) = \sum_{k=i}^{j} Var({w_k}^T X), 1 \le i \le j \le D$ *and* $Var({w_k}^T X)$ *indicates the variance of* ${w_k}^T X$.

Proof. For $\forall d \in \{1, 2, ..., D\}$, we have:

$$\begin{aligned}
&\mathbb{E}[\|{W_d}^T X_1 - {W_d}^T X_2\|_2^2] = \mathbb{E}[[{W_d}^T (X_1 - X_2)]^T [{W_d}^T (X_1 - X_2)]] \\
&= \mathbb{E}[\mathrm{tr}((X_1 - X_2)^T W_d {W_d}^T (X_1 - X_2))] \\
&= \mathbb{E}[\mathrm{tr}({W_d}^T (X_1 - X_2)(X_1 - X_2)^T W_d)] \\
&= \mathrm{tr}({W_d}^T (\mathbb{E}[X_1 {X_1}^T] + \mathbb{E}[X_2 {X_2}^T] - \mathbb{E}[X_1 {X_2}^T] - \mathbb{E}[X_2 {X_1}^T]) W_d) \\
&= \mathrm{tr}({W_d}^T (2\mathbb{E}[X X^T] - \mathbb{E}[X_1]\mathbb{E}[{X_2}^T] - \mathbb{E}[X_2]\mathbb{E}[{X_1}^T]) W_d) \\
&= 2\mathbb{E}[\mathrm{tr}({W_d}^T X X^T W_d)] = 2\mathbb{E}[X^T W_d {W_d}^T X] = 2\mathbb{E}\left[\sum_{k=1}^{d} (X^T {w_k} {w_k}^T X)\right] \\
&= 2\sum_{k=1}^{d} \mathbb{E}[({w_k}^T X)^2] = 2\sum_{k=1}^{d} \mathrm{Var}({w_k}^T X) = 2\sigma^2(1,d)
\end{aligned} \tag{3}$$

From line 1 to line 2, the inner part of the expectation is a real number. Thus, the trace operation can be safely added. From line 2 to line 3, the orders among elements are changed thanks to the nature of trace operation. From line 3 to line 4, the order of the experation and trace is changed, where the correctness stems form the addictivity of mathematical expectations. Line 5 is obtained since all $\mathbb{E}[X_i]$ are equal to 0 (see the zero-mean assumption in **Lemma 1**).

When $d = D$, we have $\mathbb{E}[\|X_1 - X_2\|_2^2] = \mathbb{E}[\|{W_D}^T (X_1 - X_2)\|_2^2]$ due to the property of orthogonal projection. Thus, we have the following equation.

$$\begin{aligned}
\mathbb{E}[\|X_1 - X_2\|_2^2] &= \mathbb{E}[\|{W_D}^T (X_1 - X_2)\|_2^2] \\
&= \frac{\sigma^2(1,D)}{\sigma^2(1,d)}\mathbb{E}[\|{W_d}^T X_1 - {W_d}^T X_2\|_2^2]
\end{aligned} \tag{4}$$

□

From above Lemma, we can see that the *exact* distance can be *unbiasedly* estimated by the variance of each dimension and the distance in projected $\mathbb{R}^d$ Euclidean space.

3.2 Optimized Estimation

Equation 4 provides a general formulation of the unbiased estimation of Euclidean distance in terms of data distribution. In this subsection, we further study the optimal estimation, which minimizes the variance of the difference between the estimated distance and the true distance. We let $\Delta X := X_1 - X_2 \in \mathbb{R}^D$. For any d, the optimal estimation can be described as an optimization problem:

$$\begin{aligned}
\min_{W_D \in \mathbb{R}^{D\times D}} \quad & \mathbb{E}\left[\left(\frac{\sigma^2(1,D)}{\sigma^2(1,d)}\|{W_d}^T \Delta X\|_2^2 - \|{W_D}^T \Delta X\|_2^2\right)^2\right] \\
\text{s.t.} \quad & {W_D}^T W_D = \mathbf{I}
\end{aligned} \tag{5}$$

It should be noted that we should only have a uniform W_D, where W_d is the first d dimension of W_D, since obtaining different W_d for each d is time-consuming to perform transformation and memory-consuming to store such plenty of transformed data objects. Thus, we propose to optimize an alternative objective as described as follows.

Lemma 4. *For any* d*, minimizing Equation 5 can be approximately achieved through maximizing* $\sigma^2(1,d)$.

Proof. For the internal part, we have:

$$\begin{aligned}
&\left(\frac{\sigma^2(1,D)}{\sigma^2(1,d)}\|{W_d}^T \Delta X\|_2^2 - \|W_D^T \Delta X\|_2^2\right)^2 \\
&= \left[\frac{\sigma^2(1,D)}{\sigma^2(1,d)}\sum_{k=1}^{d}\left(\Delta X^T {w_k}{w_k}^T \Delta X\right) - \sum_{k=1}^{D}\left(\Delta X^T {w_k}{w_k}^T \Delta X\right)\right]^2 \\
&= \left\{\sum_{k=1}^{D}\left[\left(\frac{\sigma^2(d+1,D)}{\sigma^2(1,d)}\mathbb{I}_{k\le d} - \mathbb{I}_{k>d}\right)\Delta X^T {w_k}{w_k}^T \Delta X\right]\right\}^2
\end{aligned} \tag{6}$$

where $\mathbb{I}_{k\le d}$ equals to 1 if $k \le d$, otherwise 0. We define $L_d \in \mathbb{R}^{D\times D}$ as a diagonal matrix as follows:

$$L_d := \mathrm{diag}\left(\frac{\sigma^2(d+1,D)}{\sigma^2(1,d)}, ..., \frac{\sigma^2(d+1,D)}{\sigma^2(1,d)}, -1, ..., -1\right) \tag{7}$$

where the first d elements of L_d are $\sigma^2(d+1,D)/\sigma^2(1,d)$, and the remaining in the main diagonal elements are -1. We also define $\Sigma := {W_D}^T \Delta X \Delta X^T W_D \in \mathbb{R}^{D\times D}$. With these definitions, Equation 6 can be rewritten in matrix form as follows:

$$\begin{aligned}
&\left\{\sum_{k=1}^{D}\left[\left(\frac{\sigma^2(d+1,D)}{\sigma^2(1,d)}\mathbb{I}_{k\le d} - \mathbb{I}_{k>d}\right)\Delta X^T {w_k}{w_k}^T \Delta X\right]\right\}^2 \\
&= (\Delta X^T W_D L_d {W_D}^T \Delta X)^2 = \Delta X^T W_D L_d {W_D}^T \Delta X \Delta X^T W_D L_d {W_D}^T \Delta X \\
&= \mathrm{tr}(\Delta X^T W_D L_d {W_D}^T \Delta X \Delta X^T W_D L_d {W_D}^T \Delta X) \\
&= \mathrm{tr}((({W_D}^T \Delta X)(\Delta X^T W_D L_d {W_D}^T \Delta X \Delta X^T W_D L_d)) \\
&= \mathrm{tr}(\Sigma L_d \Sigma L_d) \le \mathrm{tr}\left(\Sigma {L_d}^{\frac{1}{2}} ({L_d}^{\frac{1}{2}})^H \Sigma {L_d}^{\frac{1}{2}} ({L_d}^{\frac{1}{2}})^H\right) \\
&= \mathrm{tr}\left(({L_d}^{\frac{1}{2}})^H \Sigma {L_d}^{\frac{1}{2}} ({L_d}^{\frac{1}{2}})^H \Sigma {L_d}^{\frac{1}{2}}\right) = \|({L_d}^{\frac{1}{2}})^H \Sigma {L_d}^{\frac{1}{2}}\|_F^2 \\
&\le \|\Sigma\|_F^2 \|{L_d}^{\frac{1}{2}}\|_F^4 = \|{W_D}^T \Delta X \Delta X^T W_D\|_F^2 \|{L_d}^{\frac{1}{2}}\|_F^4 = \|\Delta X \Delta X^T\|_F^2 \|{L_d}^{\frac{1}{2}}\|_F^4
\end{aligned} \tag{8}$$

Thus, our optimization goal (c.f. Equation 5) can be approximately presented as follows.

$$\min_{W_D \in \mathbb{R}^{D\times D}} \mathbb{E}[\|\Delta X \Delta X^T\|_F^2 \|{L_d}^{\frac{1}{2}}\|_F^4] = \mathbb{E}[\|\Delta X \Delta X^T\|_F^2]\|{L_d}^{\frac{1}{2}}\|_F^4 \tag{9}$$

From this equation, we can know that this objective can be achieved through minimizing $\|{L_d}^{\frac{1}{2}}\|_F^4$. According to the definition of L_d, the final optimization is to minimize $\sigma^2(d+1,D)/\sigma^2(1,d)$. From **Lemma 2**, we can know that orthogonal projection does not change the sum of variances of all dimensions, which means $\sigma^2(d+1,D) + \sigma^2(1,d)$ is a constant. Therefore, Equation 9 is equivalent to maximize $\sigma^2(1,d)$. Proof complete. □

Thus, from this Lemma, by the definition of $\sigma^2(1,d)$, our optimization goal can be reformulated as follows:

$$\begin{aligned}
\max_{W_D \in \mathbb{R}^{D\times D}} \sigma^2(1,d) &= \sum_{k=i}^{d} \mathrm{Var}({w_k}^T X) = \mathbb{E}[X^T W_d {W_d}^T X] \\
&= \max \mathrm{tr}({W_d}^T \mathbb{E}[X X^T] W_d) \\
&\text{s.t. } {W_d}^T W_d = \mathbf{I}
\end{aligned} \tag{10}$$

According to the previous study [55], it is easy to know that this is the optimization objective of principal components analysis

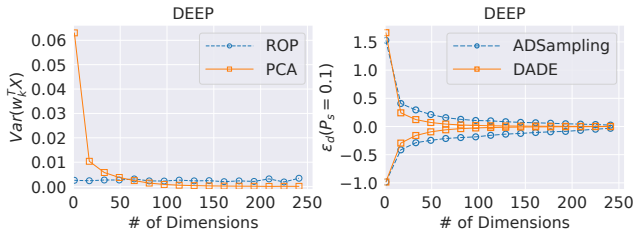

Figure 1: Running Example on DEEP.

(PCA), in which $\mathbb{E}[XX^T]$ is approximated by all data objects. Therefore, the solution for this problem can be obtained through matrix decomposition on $\mathbb{E}[XX^T]$, where λ_k is the k *largest* eigenvalue and w_k is the corresponding eigenvector.

$$\mathbb{E}[XX^T]w_k = \lambda_k w_k, \ for\ k = 1, 2, ..., d \tag{11}$$

Moreover, we can also notice the following equation hold:

$$\mathrm{Var}({w_k}^T X) = \mathbb{E}[{w_k}^T XX^T w_k] = {w_k}^T \mathbb{E}[XX^T]w_k = \lambda_k \tag{12}$$

Hence, combining Equation 4 and 12, we can know that the optimized distance estimation is as follows:

$$\mathbb{E}[\|X_1 - X_2\|_2^2] = \frac{\sum_{k=1}^{D}\lambda_k}{\sum_{k=1}^{d}\lambda_k}\mathbb{E}[\|{W_d}^T X_1 - {W_d}^T X_2\|_2^2] \tag{13}$$

It is worth mentioning that for any d, the transformation W_d obtained through PCA is optimal for Equation 10. Thus, we only need to transform once and store the transformed data objects once to make the time and space consumption acceptable. We provide an empirical study on the DEEP dataset to show the variance of each dimension in the projected space by comparing *PCA* and *randomly orthogonal projection* (ROP) as shown in the left panel of Figure 1. We can see that the random approach has almost uniform variance while *PCA* can achieve greater variance with fewer dimensions. Thus, from Equation 13, we can know that *PCA* is a more powerful approach to approximate the exact distance. It should be noted that the distance estimation of *ADSampling* [20] is *unbiased* in terms of the transformation rather than the data distribution.

3.3 Dimension Expansion with Hypothesis Testing

Projecting objects into vectors with *equal* dimensions to approximate the *exact* distance usually has two issues. First, it is knowledge-demanding and difficult to determine the number of dimensions to approximate the *exact* distance with sufficient confidence in practice. Second, conducting DCOs with *equal* dimensions has inferior accuracy since different objects may require different numbers of dimensions to make a good decision (see Section 4.2.2). Therefore, we leverage hypothesis testing to *adaptively* determine the number of dimensions when the query object q, the candidate object o, and the distance of the K-th nearest neighbor are given.

Specifically, we propose to determine the number of dimensions of distance estimation in an *incremental* manner, which is similar to the method *ADSampling*. However, different from *ADSampling* that has *unbiased* estimation in terms of the transformation matrix, in which there exists a concentration inequality on the approximate distance (see Lemma 3 in [20] for details), our method *DADE* provides an *unbiased* estimation in terms of data distribution, where

Algorithm 1: *DADE*

Input: A transformed data vector o', a transformed query vector q', a distance threshold r and the incremental step size Δ_d

Output: The results of DCO (i.e., whether $dis < r$): 1 means yes and 0 means no; When the answer is yes, the exact distance is also returned

```
Initialize the number of sampled dimensions d to be 0;
while d < D do
    d = d + Δ_d;
    Using the first d dimensions to compute the estimated
      distance dis' according to Equation 13;
    Conduct the hypothesis testing as stated in section 3.3;
    if H_0 is rejected and d < D then
        return 0;
    end
    else if H_0 is not rejected and d < D then
        Continue;
    end
    else
        return 1 and dis' if dis' ≤ r and 0 otherwise;
    end
end
```

the data distribution is unknown and has unclear expression. Thus, it is difficult to set the significance level in the hypothesis testing. To deal with this problem, we define the following probability.

$$\mathbb{P}\{\frac{\|\frac{\sum_{k=1}^{D}\lambda_k}{\sum_{k=1}^{d}\lambda_k}{W_d}^T(X_1 - X_2)\|}{\|X_1 - X_2\|} - 1 > \epsilon_d\} = P_s \tag{14}$$

where P_s is the significance level to be set as a hyper-parameters. It means a probability that the difference between the approximated distance and the exact distance is greater than ϵ_d. With the provided P_s, ϵ_d can be estimated through uniformly sampled data objects. It should be noted that ϵ_d may be different for a fixed P_s and different d. Then, we define $dis' = \|\frac{\sum_{k=1}^{D}\lambda_k}{\sum_{k=1}^{d}\lambda_k}{W_d}^T(X_1 - X_2)\|$ as the estimated distance and $dis = \|X_1 - X_2\|$ as the exact distance for presentation convenience. The hypothesis testing can be conducted as follows.
(1) We define the null hypothesis $H_0 : dis < r$ and its counterpart $H_1 : dis \geq r$.
(2) We set the significance level P_s empirically as a small value (e.g., 0.1 in our experiments), which indicates that the difference between dis' and dis is bounded by $\epsilon_d \cdot dis$ with the failure probability at most P_s.
(3) We check whether the event $dis' > (1+\epsilon_d) \cdot r$ happens. If so, we can reject H_0 and conclude $H_1 : dis > r$ with sufficient confidence since this event has a small probability, which is almost impossible to happen in one experiment.

We also provide an empirical study on DEEP dataset as shown in the right part of Figure 1, in which the x-axis indicates the number of dimensions and y-axis presents ϵ_d where $\mathbb{P}(dis'/dis - 1 > \epsilon_d) = 0.1$ for the upper two curves and $\mathbb{P}(dis'/dis - 1 < \epsilon_d) = 0.1$ for the bottom two curves. From this figure, we can have two observations: (1) *PCA* has a better approximation to the exact distance since it

has smaller deviations with the same number of dimensions; (2) compared with *random orthogonal projection*, *PCA* needs smaller dimension to reach the same significance level for the estimated distance, which means *DADE* is more efficient since it is more likely to reject H_0 compared with *ADSampling* when the significance level is fixed.

3.4 *DADE* Summarization

The process of *DADE* is summarized in Algorithm 1, which takes the transformed data and query vectors, a distance threshold r (i.e., the distance between the query and the K-th nearest neighbor), and Δ_d as inputs. It runs in an *incremental* way, which initializes the number of dimensions as 0 and increment it with Δ_d (Lines 1-3). At each loop, we first calculate the estimated distance dis' (Line 4) and then conduct the hypothesis testing (Line 5). If H_0 is rejected, we can conclude that $dis > r$ with sufficient confidence and exit the DCOs program immediately (Lines 6-7). If H_0 is not rejected, it means that we do not have enough confidence to judge whether $dis < r$. Thus, we have to continue to increment the number of dimensions to obtain more accurate dis' (Lines 9-10). For the other situation (i.e., $d = D$), the dis' will be the exact distance. Thus, we will directly compare dis' and r and return the results (Line 13).

Failure Probability Analysis. We provide the following Lemma to present the failure probability when *DADE* is adopted.

Lemma 5. *The failure probability of* DADE *is given by*

$$\begin{aligned} &\mathbb{P}\{failure\} = 0 \; if \; dis > r \\ &\mathbb{P}\{failure\} \leq \lfloor \frac{D-1}{\Delta d} \rfloor \cdot P_s \; if \; dis \leq r \end{aligned} \tag{15}$$

Proof. From Algorithm 1, it is known that *DADE* exits when $dis' > r$ or $d = D$. If $dis > r$, in these situations, our proposed *DADE* returns 0, which is always correct. Thus, $\mathbb{P}\{failure\} = 0$ can be concluded when $dis > r$. Now we consider the other situation that $dis \leq r$, which can be verified as follows.

$$\begin{aligned} &\mathbb{P}\{failure\} = \mathbb{P}\{\exists d < D, dis' > (1+\epsilon_d) \cdot r\} \\ &\leq \sum_{d=1}^{\lfloor (D-1)/\Delta d \rfloor} \mathbb{P}\{dis' > (\epsilon_d + 1) \cdot dis\} \leq \sum_{d=1}^{\lfloor (D-1)/\Delta d \rfloor} P_s \end{aligned} \tag{16}$$

where the first equation holds since a failure happens if and only if we reject H_0 for some $d < D$. □

Recall that DCO is ubiquitous in almost all AKNN algorithms. For example, for a graph-based method such as HNSW, *greedy beam search* [51] is conducted at layer 0, which is also adopted by most graph-based approaches [18, 28, 39]. It maintains two sets, i.e., a search set $\mathcal{S}$ and a result set $\mathcal{R}$, where the size of $\mathcal{S}$ is unbounded to store the candidates yet to be searched and the size of $\mathcal{R}$ is bounded by N_{ef} to maintains the N_{ef} nearest neighbors visited so far. At each iteration, it pops the object with the smallest distance in $\mathcal{S}$ and enumerates its neighbors. For each neighbor, it conducts DCO to check whether its distance to query is no greater than the maximum distance in $\mathcal{R}$. If so, it pushes this object into $\mathcal{R}$ and $\mathcal{S}$, in which the object with maximum distance in $\mathcal{R}$ will be removed whenever $\mathcal{R}$ is full. For a quantization-based method such as *IVF*, it first selects the N_{probe} nearest clusters based on the distance from the query to its centroids. Then, it scans all candidates, in which it maintains a KNN set $\mathcal{K}$ with a max-heap of size K [20]. For each one, it conducts DCO to check whether its distance to query is no greater than the maximum distance in $\mathcal{K}$. If so, it updates $\mathcal{K}$ with the scanned object.

Table 1: Dataset Statistics

Dataset	Cardinality	Dimension	Query Size	Data Type
MSong	992,272	420	200	Audio
DEEP	1,000,000	256	1,000	Image
Word2Vec	1,000,000	300	1,000	Text
GIST	1,000,000	960	1,000	Image
GloVe	2,196,017	300	1,000	Text
Tiny5M	5,000,000	384	1,000	Image

4 EXPERIMENT

4.1 Experimental Settings

Datasets. We conduct our experiments on six public datasets with different cardinalities and dimensionalities to be in line with various benchmark AKNN algorithms [20, 32, 37]. The dataset statistics are shown in Table 1.

Algorithms. For DCOs, We compare our method *DADE* with the conventional method, i.e., *FDScanning*, and the **SOTA** approach, i.e., *ADSampling* [20]. The other distance estimation techniques such as Product Quantization [29] is ignored in our experiments since *ADSampling* has empirically demonstrate a superior performance compared with them [20].

- *FDScanning*: compute the *exact* distance dis with full D dimensions, and then determine whether $dis < r$.
- *ADSamping*: estimate the distance in the low dimension space with *randomly orthogonal transformation*, in which the number of sampled dimensions *adaptively* evaluated, and then determine whether $dis < r$ with the approximated distance dis'.

We combine each method of DCOs above with classical AKNN search algorithms, e.g., *IVF* and *HNSW*, and define a set of competitors as follows.

- *HNSW* [39]: the vanilla hierachical navigable small world graph, in which *FDSanning* is adopted as DCOs.
- *HNSW+* [20]: *HNSW* with *ADSampling* as DCOs.
- *HNSW++* [20]: *HNSW* with optimizing through decoupling the roles of candidate list, i.e., one for providing the distance threshold for DCOs and one for maintaining the searched objects, in which *ADSamping* is adopted as DCOs.
- *HNSW**: *HNSW* with our proposed *DADE* as DCOs.
- *HNSW***: *HNSW++* with our proposed *DADE* as DCOs.

Similarly, we define a set of variants of *IVF*, where *IVF++* [20] is the cache-friendly version. We refer the readers to the seminal work [20] for more details about the variants.

Performance Metrics. We use Recall [20, 21, 52, 53], i.e., the overlap ratio between the results returned by AKNN algorithms and the ground-truths. We adopt query-per-second (QPS) [20, 37], i.e., the number of handled queries per second, to measure efficiency. It should be noted that for AKNN search, greater QPS with the same recall indicates a better algorithm.

Implementations. We implement the *HNSW*-related approaches such as *HNSW*** based on hnswlib [39] and implement *IVF*-related approaches such as *IVF*** based on Faiss [30] library. We obtain the transformation matrix for *ADSampling* and *DADE* by using NumPy library. In the query phase, all algorithms are implemented

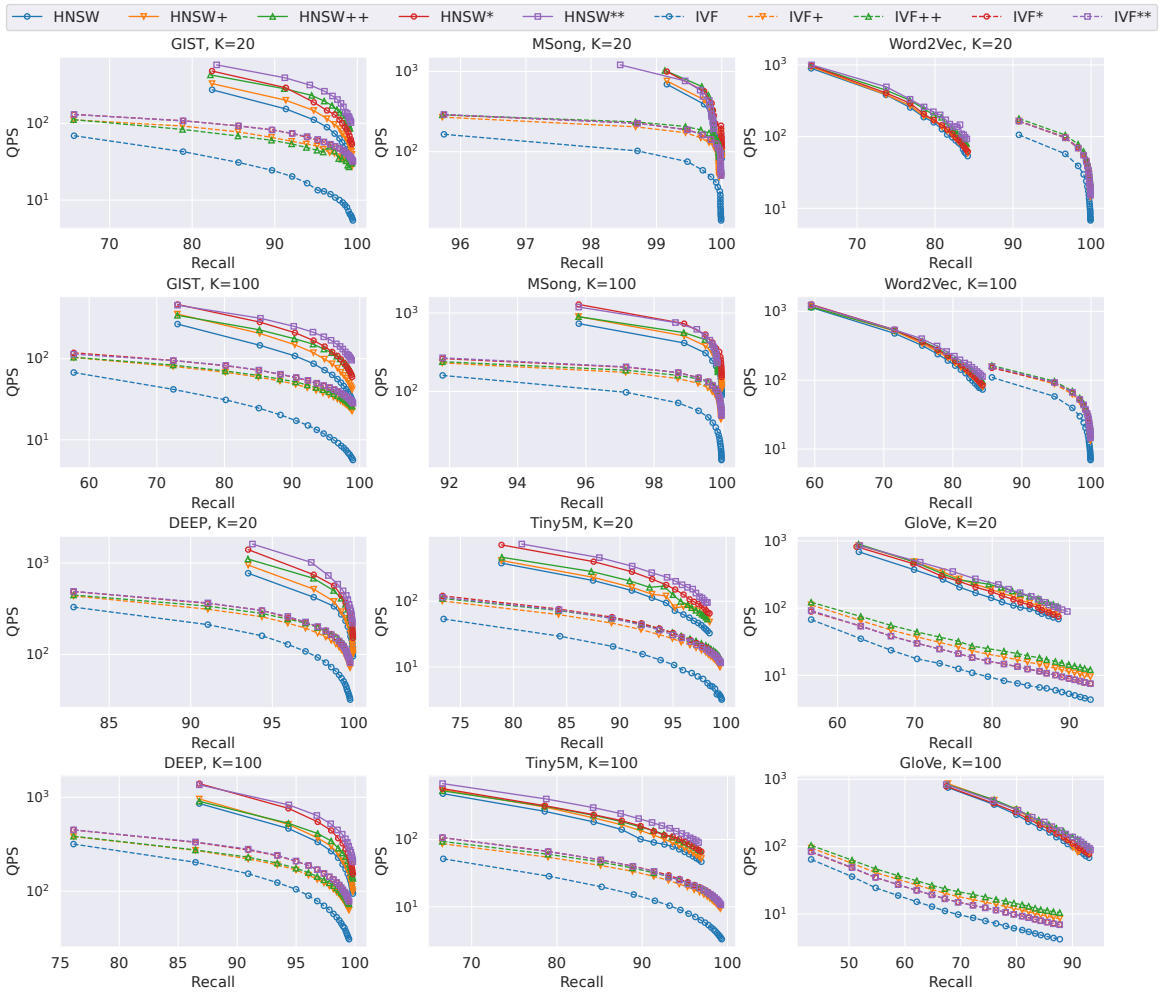

Figure 2: Time-Recall Tradeoff.

with C++. Following previous studies [20, 51], all hardware-specific optimizations including SIMD, and multi-threading are prohibited for a fair comparison. All C++ codes are complied by g++ 7.5.0 with $-O3$ optimization and run in the platform with Ubuntu 16.04 operating system with 48-cores Intel(R) CPU E5-2650 v4 @ 2.20GHz 256GB RAM.

Parameter Setting. For all *HNSW*-related approaches, two hyper-parameters are empirically preset, i.e., the number of connected neighbors M and the maximum size of the results set $efConstruction$. Following previous studies [20], we set M and $efConstruction$ to 16 and 500 respectively. For *IVF*-related approaches, the number of clusters N_c is the critical hyper-parameters in the index phase. Following Faiss [30], we set N_c to be around the square root of the cardinality, i.e., 4096 in our experiments. For *ADSampling*, we use their default parameters, i.e., $\epsilon_0 = 2.1$. For *DADE*, we empirically set P_s to 0.1 and the step size of dimension expansion Δ_d to 32.

4.2 Experimental Results

4.2.1 ***Overall Performance.*** We report the experimental results in terms of Recall and QPS in Figure 2. Specifically, we vary N_{ef} (i.e., the maximum size of the result set $\mathcal{R}$) from 100 to 1500 with step size 100 for the *HNSW*-related approaches, and vary N_{nprobe} (i.e., the number of clusters to be searched) from 20 to 400 with step size 20 for *IVF*-related approaches to show the trade-off between time and accuracy. From the results, we have the following observations. (1) From the index perspective, *HNSW* outperforms *IVF* in most cases. The techniques such as decoupling the roles of the candidate list for *HNSW* and cache-friendly optimization for *IVF* proposed in [20] are effective, which clearly improves the efficiency without affecting the accuracy (e.g., *HNSW*** outperform *HNSW** with a large margin in most cases). (2) From the DCO perspective, *DADE* and *ADSampling* can achieve better trade-off compared with *FDScanning* (e.g., *IVF** and *IVF+* outperform the vanilla *IVF*). Moreover, compared with the best-performing competitor *ADSampling*, our proposed method *DADE* consistently improves the efficiency with a large margin. Recall that the only difference between the methods such as *IVF** and *IVF+* lies in its method of DCO. For example, on the DEEP dataset with $K = 100$, *HNSW** achieves the recall of 86% with QPS of 140, while *HNSW+* has little decline of recall with QPS of 96. In this situation, our method improves the efficiency by more than 45%. When focusing on the high accuracy region such as 99% recall on the DEEP dataset, our method even provides a better improvement. For example, *HNSW∗* achieves a recall of 99.2 with a QPS of 28, while *HNSW+* achieves the same recall with a QPS of 18.

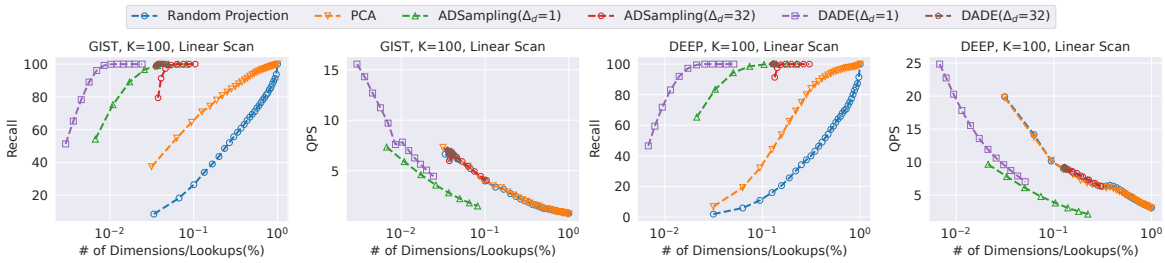

Figure 3: Feasibility for DCOs in terms of Recall and QPS.

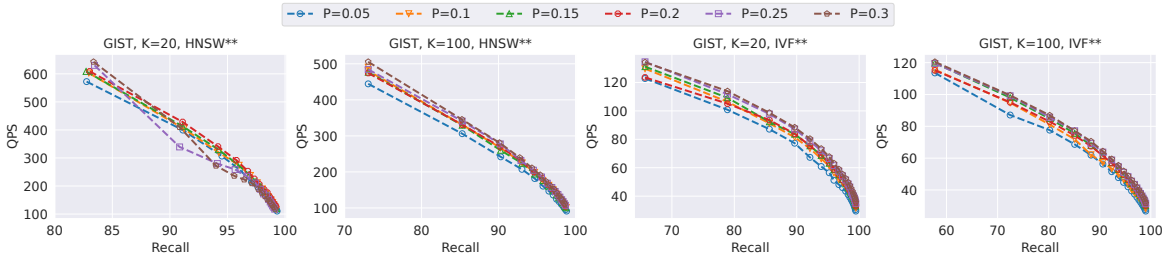

Figure 4: Parameter Study on p of AKNN Algorithms with Different K.**

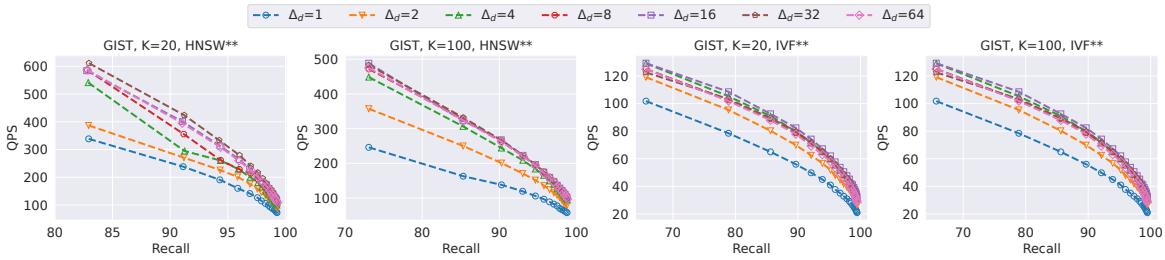

Figure 5: Parameter Study on Δ_d of AKNN Algorithms with Different K.**

4.2.2 ***Feasibility of Distance Estimation Methods for DCOs.*** Next, we study the feasibility of various distance estimations, i.e., *Random Projection*, *PCA*, *ADSampling*, and *DADE*. To eliminate the effect of different index structures, we follow [20] to conduct this experiment with an exact KNN algorithm, called *Linear Scan*. Specifically, we scan all the data objects and return the K nearest neighbors to the queries. For *ADSampling* and *DADE*, we maintain a KNN set like *IVF* and conduct DCOs for each object sequentially. We plot the curves of recall-number of dimensions and QPS-number of dimensions in Figure 3, where the value of x-axis can be computed through the number of dimensions used for distance calculation in different DCO methods divides the total dimensions when *FDScanning* is adopted. For *random projection* and *PCA*, we vary the dimensionality of the projected vectors (i.e., the estimated distance is based on *equal* dimensions rather than *adaptively* set in query phase).

From the results, we can observe that (1) when reducing the number of dimensions to 0.1, the recall of *random projection* will no more than 40%; (2) when *PCA* is adopted for distance estimation, the recall has significant improvement compared with *random projeciton*, which demonstrates the necessity to perform data-aware distance estimation; (3) *ADSampling* and *DADE* only need less than 0.1 dimensions on average to achieve more than 90% recall, which may demonstrate that for an object that is far away from the query, we only need few dimensions to correctly confirm $dis > r$. Thus, the *exact* distance computation in DCOs is unnecessary; (4) For *ADSampling* and *DADE*, the curves are plotted through varying ϵ_0 from 0.5 to 4.0 and varying P_s from 0.05 to 0.6. Compared with *ADSampling*, *DADE* achieves better recall when the number of dimensions equals, which confirms the results from Section 4.2.1; (5) compared with *PCA*, *DADE* that adopts hypothesis testing to *adaptively* determine the number of dimensions to approximate the distance, achieves better performance in terms of recall and QPS since different candidates may require different number of dimensions to check whether $dis < r$ with sufficient confidence. This observation also shows the effectiveness of our proposed hypothesis testing method.

4.2.3 ***Sensitivity Study.*** We study two hyper-parameters in *DADE*, i.e., the significance level P_s in hypothesis testing and the step size Δ_d in the dimension expansion, where these factors control the trade-off between accuracy and QPS. In the following section, for a given P_s and Δ_d, each curve is plotted through varying N_{ef} and N_{probe} for *HNSW*** and *IVF***, respectively.

The effect of P_s. We conduct the experiment by varying P_s from 0.05 to 0.3 with step size 0.05. It should be noted that the smaller P_s is, the more accurate the estimated distance is. The results are

shown in Figure 4. We can observe that the curve moves to the upper right with the increase P_s at the beginning. With the further increase such as from $P_s = 0.25$ to $P_s = 0.3$ in the GIST dataset with $K = 100$, the curves move in an opposite way, which shows a trade-off between the estimated accuracy and efficiency. Specifically, the greater P_s is, the higher probability H_0 can be rejected (see Section 3.3). Thus, *DADE* will exit the loop earlier with a lower number of dimensions, which will improve the efficiency. However, the increases of P_s will cause many more failures (i.e., H_0 is rejected with $dis < r$). Therefore, when P_s excels at a specific threshold, the increase in failure will degrade the performance of AKNN search algorithms.

The effect of Δ_d. We conduct the experiment by varying Δ_d from 1 to 64 with uneven intervals. The results are shown in Figure 5. From this figure, we have the following observations. (1) With the increase of Δ_d, the curves move from the bottom left to the upper right first. Then, the curves move to the opposite way. This is because when Δ_d is small such 1, *DADE* will cost time to increase the number of dimensions to have sufficient confidence to check whether H_0 will be rejected. Although smaller Δ_d has the ability to quit the DCO procedure with lower dimensions, hypothesis testing may be time-consuming. When Δ_d is large enough such as 32, *DADE* can increase the number of dimensions with fewer loops, which decreases the times of hypothesis testing and achieves the best trade-off. (2) The search through *Linear Scan* and *HNSW*** (or *IVF***) shows different preferences. For example, when *Linear Scan* with *DADE* is adopted, $\Delta_d = 1$ achieves the best performance (see Figure 3). This is because *Linear Scan* treats all data objects as the candidates, in which most of them can be eliminated with high probability when the number of dimensions is lower since the distance between the object and query is far away.

5 RELATED WORKS

Approximate K Nearest Neighbor Search. Existing approaches for AKNN search can be roughly divided into (1) graph-based [3, 18, 39, 43, 52], (2) quantization-based [4, 21, 22, 29], (3) tree-based [7, 12, 14, 40] and (4) hashing-based [15, 19, 46, 53]. Different categories of these methods show different advantages. For example, graph-based methods usually achieve the best performance for both in-memory [37, 52] and disk-resident situations [10, 28, 50]. Quantization-based methods outperform the others in terms of memory consumption. Hash-based methods provide a theoretical guarantee in terms of the quality of the searched objects. Beyond these methods, there are various studies that apply machine learning techniques to AKNN search [24, 38, 44, 54]. For example, BLISS [24] adopts multilayer perceptron to predict the bucket id of each data object and performs repartition to make the objects in each bucket more compact in terms of KNN. Learning to hashing [49] adopts the metric learning technique to group the similar data objects into the same bucket. Zheng et.al [54] propose to learn the number of buckets to be scanned for each query in *IVF* index to reduce the number of computations in total since different queries require different numbers of buckets to achieve the same accuracy. Although these approaches provide different ways to generate the candidates, the methods for DCOs are orthogonal to these approaches, which focus on finding KNNs among the generated candidates.

Distance Estimation. Random Projection is a well-known technique to approximate the Euclidean distance, which is widely used in LSH [46, 53]. For example, Zheng et.al [53] propose PM-LSH to perform query-aware hashing for objects, in which the *exact* Euclidean distance *dis* can be estimated in the projected space with $dis'/\sqrt{m}$, where m is the dimension of the projected space. However, it should be noted that the random projection used in LSH is for generating candidates rather than conducting DCOs. Arora et al. [1] propose HD-Index, where the upper-bound of the Euclidean distance is fast estimated through triangular and Ptolemaic inequality to refine candidates. Similarly, Li et al. [33] adopt transformation with PCA to obtain a tighter upper-bound when inner product distance is applied. Recently, Gao et.al [20] propose *ADSampling* that leverages the *random orthogonal projection* for distance estimation in DCOs, i.e., $\sqrt{D/d} \cdot dis'$. According to our best knowledge, this is the only work that focuses on the speed-up of DCOs. However, the distance estimation in *ADSampling* is data-oblivious, which hinders the accuracy of the distance estimation for a specific dataset. Moreover, the distance estimation in *ADSampling* is *unbiased* in terms of the random projection rather than the data distribution since the explicit expression of the data distribution is *unknown*, which makes it hard to develop a theory to provide guarantee about the distance estimation. To bridge this gap, we propose a general formation of the *unbiased* estimation in terms of data distribution and theoretically show an optimized approach for more accurate distance estimation.

6 CONCLUSION

This paper proposes a data-aware distance estimation, called *DADE*, to speed up the process of DCOs, which aims to return KNNs from the candidates set. Specifically, *DADE* first rotates the original space with an *orthogonal transformation* W, where the variance of each dimension in the projected space is ordered from large to small. Then, it approximates the *exact* distance in the space with lower dimensions, where the number of dimensions is *adaptively* determined in the query phase through a hypothesis testing approach. Moreover, the probability that controls the significance level is defined and empirically approximated from the data objects in the hypothesis testing. We theoretically prove that the distance estimation in *DADE* is *unbiased* and *optimized* in terms of data distribution if the transformation is *orthogonal*. We conduct extensive experiments on widely used benchmark datasets compared with conventional and SOTA methods for DCOs by combining different index structures. The results demonstrate that our proposed method *DADE* can outperform existing DCO methods to achieve a better trade-off between accuracy and latency.

ACKNOWLEDGMENTS

This work is partially supported by NSFC (No. 62472068), Shenzhen Municipal Science and Technology R&D Funding Basic Research Program (JCYJ20210324133607021), and Municipal Government of Quzhou under Grant (No. 2023D044), and Key Laboratory of Data Intelligence and Cognitive Computing, Longhua District, Shenzhen.

REFERENCES

[1] Akhil Arora, Sakshi Sinha, Piyush Kumar, and Arnab Bhattacharya. 2018. HD-Index: Pushing the Scalability-Accuracy Boundary for Approximate kNN Search in High-Dimensional Spaces. *Proceedings of the VLDB Endowment* 11, 8 (2018).

[2] Akari Asai, Sewon Min, Zexuan Zhong, and Danqi Chen. 2023. Tutorial Proposal: Retrieval-based Language Models and Applications. In *The 61st Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*. 41.

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

[4] Artem Babenko and Victor Lempitsky. 2014. The inverted multi-index. *IEEE transactions on pattern analysis and machine intelligence* 37, 6 (2014), 1247–1260.

[5] Norbert Beckmann, Hans-Peter Kriegel, Ralf Schneider, and Bernhard Seeger. 1990. The R*-tree: An efficient and robust access method for points and rectangles. In *Proceedings of the 1990 ACM SIGMOD international conference on Management of data*. 322–331.

[6] Jon Louis Bentley. 1975. Multidimensional binary search trees used for associative searching. *Commun. ACM* 18, 9 (1975), 509–517.

[7] Alina Beygelzimer, Sham Kakade, and John Langford. 2006. Cover trees for nearest neighbor. In *Proceedings of the 23rd international conference on Machine learning*. 97–104.

[8] Jin Chen, Zheng Liu, Xu Huang, Chenwang Wu, Qi Liu, Gangwei Jiang, Yuanhao Pu, Yuxuan Lei, Xiaolong Chen, Xingmei Wang, et al. 2024. When large language models meet personalization: Perspectives of challenges and opportunities. *World Wide Web* 27, 4 (2024), 42.

[9] Patrick Chen, Wei-Cheng Chang, Jyun-Yu Jiang, Hsiang-Fu Yu, Inderjit Dhillon, and Cho-Jui Hsieh. 2023. Finger: Fast inference for graph-based approximate nearest neighbor search. In *Proceedings of the ACM Web Conference 2023*. 3225–3235.

[10] Qi Chen, Bing Zhao, Haidong Wang, Mingqin Li, Chuanjie Liu, Zengzhong Li, Mao Yang, and Jingdong Wang. 2021. Spann: Highly-efficient billion-scale approximate nearest neighborhood search. *Advances in Neural Information Processing Systems* 34 (2021), 5199–5212.

[11] Krzysztof M Choromanski, Mark Rowland, and Adrian Weller. 2017. The unreasonable effectiveness of structured random orthogonal embeddings. *Advances in neural information processing systems* 30 (2017).

[12] Paolo Ciaccia, Marco Patella, Pavel Zezula, et al. 1997. M-tree: An efficient access method for similarity search in metric spaces. In *Vldb*, Vol. 97. Citeseer, 426–435.

[13] Thomas Cover and Peter Hart. 1967. Nearest neighbor pattern classification. *IEEE transactions on information theory* 13, 1 (1967), 21–27.

[14] Sanjoy Dasgupta and Yoav Freund. 2008. Random projection trees and low dimensional manifolds. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*. 537–546.

[15] Mayur Datar, Nicole Immorlica, Piotr Indyk, and Vahab S Mirrokni. 2004. Locality-sensitive hashing scheme based on p-stable distributions. In *Proceedings of the twentieth annual symposium on Computational geometry*. 253–262.

[16] Liwei Deng, Hao Sun, Rui Sun, Yan Zhao, and Han Su. 2022. Efficient and effective similar subtrajectory search: a spatial-aware comprehension approach. *TIST* 13, 3 (2022), 1–22.

[17] Liwei Deng, Yan Zhao, Jin Chen, Shuncheng Liu, Yuyang Xia, and Kai Zheng. 2024. Learning to Hash for Trajectory Similarity Computation and Search. In *ICDE*. IEEE, 4491–4503.

[18] Cong Fu, Chao Xiang, Changxu Wang, and Deng Cai. 2019. Fast Approximate Nearest Neighbor Search With The Navigating Spreading-out Graph. *Proceedings of the VLDB Endowment* 12, 5 (2019).

[19] Junhao Gan, Jianlin Feng, Qiong Fang, and Wilfred Ng. 2012. Locality-sensitive hashing scheme based on dynamic collision counting. In *Proceedings of the 2012 ACM SIGMOD international conference on management of data*. 541–552.

[20] Jianyang Gao and Cheng Long. 2023. High-dimensional approximate nearest neighbor search: with reliable and efficient distance comparison operations. *Proceedings of the ACM on Management of Data* 1, 2 (2023), 1–27.

[21] Jianyang Gao and Cheng Long. 2024. RaBitQ: Quantizing High-Dimensional Vectors with a Theoretical Error Bound for Approximate Nearest Neighbor Search. *Proceedings of the ACM on Management of Data* 2, 3 (2024), 1–27.

[22] Tiezheng Ge, Kaiming He, Qifa Ke, and Jian Sun. 2013. Optimized product quantization. *IEEE transactions on pattern analysis and machine intelligence* 36, 4 (2013), 744–755.

[23] Mihajlo Grbovic and Haibin Cheng. 2018. Real-time personalization using embeddings for search ranking at airbnb. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 311–320.

[24] Gaurav Gupta, Tharun Medini, Anshumali Shrivastava, and Alexander J Smola. 2022. Bliss: A billion scale index using iterative re-partitioning. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 486–495.

[25] Antonin Guttman. 1984. R-trees: A dynamic index structure for spatial searching. In *Proceedings of the 1984 ACM SIGMOD international conference on Management of data*. 47–57.

[26] Jui-Ting Huang, Ashish Sharma, Shuying Sun, Li Xia, David Zhang, Philip Pronin, Janani Padmanabhan, Giuseppe Ottaviano, and Linjun Yang. 2020. Embedding-based retrieval in facebook search. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2553–2561.

[27] Masajiro Iwasaki and Daisuke Miyazaki. 2018. Optimization of indexing based on k-nearest neighbor graph for proximity search in high-dimensional data. *arXiv preprint arXiv:1810.07355* (2018).

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

[29] Herve Jegou, Matthijs Douze, and Cordelia Schmid. 2010. Product quantization for nearest neighbor search. *IEEE transactions on pattern analysis and machine intelligence* 33, 1 (2010), 117–128.

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

[31] Yannis Kalantidis and Yannis Avrithis. 2014. Locally optimized product quantization for approximate nearest neighbor search. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2321–2328.

[32] Conglong Li, Minjia Zhang, David G Andersen, and Yuxiong He. 2020. Improving approximate nearest neighbor search through learned adaptive early termination. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*. 2539–2554.

[33] Hui Li, Tsz Nam Chan, Man Lung Yiu, and Nikos Mamoulis. 2017. FEXIPRO: fast and exact inner product retrieval in recommender systems. In *Proceedings of the 2017 ACM International Conference on Management of Data*. 835–850.

[34] Nan Li, Bo Kang, and Tijl De Bie. 2023. SkillGPT: a RESTful API service for skill extraction and standardization using a Large Language Model. *arXiv preprint arXiv:2304.11060* (2023).

[35] Defu Lian, Yongji Wu, Yong Ge, Xing Xie, and Enhong Chen. 2020. Geography-aware sequential location recommendation. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*. 2009–2019.

[36] Yingfan Liu, Hong Cheng, and Jiangtao Cui. 2017. PQBF: i/o-efficient approximate nearest neighbor search by product quantization. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. 667–676.

[37] Kejing Lu, Mineichi Kudo, Chuan Xiao, and Yoshiharu Ishikawa. 2021. HVS: hierarchical graph structure based on voronoi diagrams for solving approximate nearest neighbor search. *Proceedings of the VLDB Endowment* 15, 2 (2021), 246–258.

[38] Zepu Lu, Jin Chen, Defu Lian, Zaixi Zhang, Yong Ge, and Enhong Chen. 2024. Knowledge distillation for high dimensional search index. *Advances in Neural Information Processing Systems* 36 (2024).

[39] Yu A Malkov and Dmitry A Yashunin. 2018. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE transactions on pattern analysis and machine intelligence* 42, 4 (2018), 824–836.

[40] Marius Muja and David G Lowe. 2014. Scalable nearest neighbor algorithms for high dimensional data. *IEEE transactions on pattern analysis and machine intelligence* 36, 11 (2014), 2227–2240.

[41] Javier Vargas Munoz, Marcos A Gonçalves, Zanoni Dias, and Ricardo da S Torres. 2019. Hierarchical clustering-based graphs for large scale approximate nearest neighbor search. *Pattern Recognition* 96 (2019), 106970.

[42] Shumpei Okura, Yukihiro Tagami, Shingo Ono, and Akira Tajima. 2017. Embedding-based news recommendation for millions of users. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. 1933–1942.

[43] Yun Peng, Byron Choi, Tsz Nam Chan, Jianye Yang, and Jianliang Xu. 2023. Efficient approximate nearest neighbor search in multi-dimensional databases. *Proceedings of the ACM on Management of Data* 1, 1 (2023), 1–27.

[44] Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan Hulikal Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Tran, Jonah Samost, et al. 2024. Recommender systems with generative retrieval. *Advances in Neural Information Processing Systems* 36 (2024).

[45] Yifang Sun, Wei Wang, Jianbin Qin, Ying Zhang, and Xuemin Lin. 2014. SRS: solving c-approximate nearest neighbor queries in high dimensional euclidean space with a tiny index. *Proceedings of the VLDB Endowment* (2014).

[46] Yao Tian, Xi Zhao, and Xiaofang Zhou. 2023. DB-LSH 2.0: Locality-sensitive hashing with query-based dynamic bucketing. *IEEE Transactions on Knowledge and Data Engineering* (2023).

[47] Ingo Wald and Vlastimil Havran. 2006. On building fast kd-trees for ray tracing, and on doing that in O (N log N). In *2006 IEEE Symposium on Interactive Ray Tracing*. IEEE, 61–69.

[48] Hao Wang, Tong Xu, Qi Liu, Defu Lian, Enhong Chen, Dongfang Du, Han Wu, and Wen Su. 2019. MCNE: An end-to-end framework for learning multiple conditional network representations of social network. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 1064–1072.

[49] Jingdong Wang, Ting Zhang, Nicu Sebe, Heng Tao Shen, et al. 2017. A survey on learning to hash. *IEEE transactions on pattern analysis and machine intelligence*

40, 4 (2017), 769–790.

[50] Mengzhao Wang, Weizhi Xu, Xiaomeng Yi, Songlin Wu, Zhangyang Peng, Xiangyu Ke, Yunjun Gao, Xiaoliang Xu, Rentong Guo, and Charles Xie. 2024. Starling: An I/O-Efficient Disk-Resident Graph Index Framework for High-Dimensional Vector Similarity Search on Data Segment. *Proceedings of the ACM on Management of Data* 2, 1 (2024), 1–27.

[51] Mengzhao Wang, Xiaoliang Xu, Qiang Yue, and Yuxiang Wang. 2021. A comprehensive survey and experimental comparison of graph-based approximate nearest neighbor search. *Proceedings of the VLDB Endowment* 14, 11 (2021), 1964–1978.

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

[53] Bolong Zheng, Zhao Xi, Lianggui Weng, Nguyen Quoc Viet Hung, Hang Liu, and Christian S Jensen. 2020. PM-LSH: A fast and accurate LSH framework for high-dimensional approximate NN search. *Proceedings of the VLDB Endowment* 13, 5 (2020), 643–655.

[54] Bolong Zheng, Ziyang Yue, Qi Hu, Xiaomeng Yi, Xiaofan Luan, Charles Xie, Xiaofang Zhou, and Christian S Jensen. 2023. Learned probing cardinality estimation for high-dimensional approximate NN search. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*. IEEE, 3209–3221.

[55] Zhi-Hua Zhou. 2021. *Machine learning*. Springer nature.