Deterministic Approximate Nearest Neighbors in High Dimensions

Ying Feng
MIT

Piotr Indyk
MIT

November 7, 2026

Abstract

In the $(1+\varepsilon)$-approximate nearest neighbor problem, we are given a set P of n points in $\mathbb{R}^d$, and the goal is to build a data structure that, for any query $q \in \mathbb{R}^d$, returns a point of P whose Euclidean distance from q is within a factor $1+\varepsilon$ of the distance from q to its nearest neighbor in P. Since the late 1990s, randomized data structures have achieved space polynomial in n and d and query time polynomial in d, $\log n$, and $1/\varepsilon$, for any constant $\varepsilon > 0$. However, their guarantees hold only with high probability, over random choices made during preprocessing, during queries, or both. Whether randomness is necessary to achieve such results has remained open since 2000: the only known deterministic data structures with comparable bounds achieve an approximation factor of $3+\varepsilon$.

In this paper we resolve this question. For every constant $\varepsilon > 0$, we give a deterministic data structure with space polynomial in n and d that answers every query correctly, with query time $O(d^2 + d\,\mathrm{poly}(\varepsilon^{-1}, \log n))$. Our construction relies on deterministic "superblock" decompositions, and its analysis replaces the triangle-inequality argument behind the factor of 3 with a second-moment argument that averages over many local data structures.

1 Introduction

Given a set P of n points in $\mathbb{R}^d$, the *nearest neighbor* problem asks for a data structure that, on input a query $q \in \mathbb{R}^d$, finds some $x \in P$ that minimizes $\|q - x\|_2 \le r$. It is one of the most basic primitives in computational geometry, and it underlies similarity search over large collections of images, documents, and other high-dimensional data [HPIM12]. When d is small, the problem admits fast exact solutions (see, e.g., [Mei93, Cla88]), but their space or query time grows exponentially with d, a phenomenon known as the *curse of dimensionality*. A way around this obstacle, developed since the late 1990s, is to relax the problem. In the $(1+\varepsilon)$-approximate version ($(1+\varepsilon)$-ANN), we are allowed to return any point $\widehat{x} \in P$ satisfying

$$\|q - \widehat{x}\|_2 \le (1+\varepsilon) \min_{x \in P} \|q - x\|_2 , \tag{1}$$

where $\varepsilon > 0$ and all distances are Euclidean[1]. This relaxation admits data structures whose complexity is polynomial in d, for every fixed $\varepsilon > 0$.

The known solutions for ANN trade the approximation factor off against space and query time. We focus on the regime of very fast queries: query time $\mathrm{poly}(d, \log n, 1/\varepsilon)$, with space polynomial in n and d whenever ε is a constant. This regime was already the subject of two works [IM98, KOR00]. Both use roughly $n^{O(1/\varepsilon)^2}$ space, but they differ in their query time and in where their randomness enters:

[1] In this paper, for simplicity, we focus on the Euclidean case. However, it is immediate that a $(1+\varepsilon)$-ANN data structure for the ℓ_2 norm also serves as a $(1+\varepsilon)^2$-ANN data structure for the Hamming distance with the same complexity. Using standard reductions [Ahl17] this also yields a $((1+\varepsilon)^2+\varepsilon)$-ANN data structure for the ℓ_1 norm.

- In [IM98], queries take $d \cdot \mathrm{poly}(\log n, 1/\varepsilon)$ time. The random choices are made during preprocessing, and the guarantee holds per query: any fixed q is answered correctly with probability $1 - 1/\mathrm{poly}(n)$ over these choices.
- In [KOR00], queries take $d^2 \cdot \mathrm{poly}(\log n, 1/\varepsilon)$ time. With high probability, preprocessing yields a structure that is good for all queries simultaneously; the query algorithm is itself randomized, and answers each q correctly with probability $1 - 1/\mathrm{poly}(n)$ over its own coin tosses.

Subsequent work has improved these bounds, e.g., reducing the query time to $O(d \log d + \varepsilon^{-3} \log^2 n)$ [AC09], and has developed data structures with many other tradeoffs between space and query time [AI06, AR15, ALRW17]; see [HPIM12, AIR18] for an overview. However, all of these data structures are randomized. This leads to a natural question: is there a *deterministic* $\mathrm{poly}(n, d)$-size data structure for $(1+\varepsilon)$-ANN that guarantees $\mathrm{poly}(d, \log n, 1/\varepsilon)$ query time, for every constant $\varepsilon > 0$?

This question has remained open since 2000, and only partial results are known (see [AIR18, Section 3]). As observed already in [KOR00], one can construct a data structure that, with high probability, is correct for all queries simultaneously; however, the construction still fails with nonzero probability, and no deterministic polynomial-time algorithm for constructing such a data structure was known. For larger approximation factors, Indyk [Ind00] gave the first deterministic polynomial-time construction of a data structure that is correct for all queries, achieving approximation factor $3 + \varepsilon$ (for the Hamming space or the ℓ_1 norm) with polynomial space and $\mathrm{poly}(d, \log n, 1/\varepsilon)$ query time.

As noted in [AIR18], this was the only known polynomial-size deterministic approximate near neighbor data structure with a constant approximation factor. More progress has been made on an intermediate problem, in which the data structure either returns a correct answer or reports failure, the latter with small probability. Indyk [Ind00] gave such a data structure for $(1+\varepsilon)$-ANN in Hamming space with bounds similar to those of [KOR00], and Pagh [Pag16], Ahle [Ahl17], and Wei [Wei22] gave such data structures with sublinear query time. In particular, Ahle matched the performance of locality-sensitive hashing in the Hamming space [IM98], and Wei matched the best known bounds for ℓ_p, $1 \le p \le 2$, including those based on data-dependent hashing [AR15]. However, such "no false negatives" guarantees are substantially weaker than deterministic ones.

1.1 Our result

Our main result is the following.

Theorem 1.1. *For every integer $d \ge 1$, every $\varepsilon \in (0,1)$, and every nonempty finite set $P \subseteq \mathbb{R}^d$ of size $n \ge d$, there is a fully deterministic data structure for $(1+\varepsilon)$-approximate nearest-neighbor search on P. Its preprocessing time and space are*

$$n^{\mathrm{poly}(\varepsilon^{-1})},$$

and its deterministic query time is

$$O\big(d^2 + d\,\mathrm{poly}(\varepsilon^{-1}, \log n)\big).$$

Organization. Section 2 outlines the construction and its analysis. After the preliminaries, Section 4 describes the data structure and derives the main theorem from two ingredients. Section 5 proves the approximation guarantee, and Section 6 gives the deterministic block construction. Appendix A supplies the deterministic implementation of the low-dimensional primitive.

AI disclosure: We used ChatGPT, Claude and Gemini to assist with drafting, revising, and checking the exposition and mathematical arguments. The tools materially affected all sections. The starting point of this work was a query to ChatGPT 5.6 Sol asking it to re-analyze the $(3+\varepsilon)$-approximation algorithm of [Ind00] in ℓ_2 using a stronger form of the triangle inequality; this yielded a $(\sqrt{5}+\varepsilon)$-approximation algorithm. The resulting proof was then provided to ChatGPT, Claude, and Gemini, with additional papers in the context, which yielded the main result of this paper. The authors verified the correctness and originality of all content, including references.

2 Technical overview

We now describe the construction and the main ideas of its analysis. Throughout the overview, all norms are Euclidean. We start by recalling the block-based approach behind the deterministic $(3+\varepsilon)$-approximation of Indyk [Ind00], which is the point of departure for our work, and explain where its factor of 3 comes from. We then explain our ideas through a simpler construction that gives $d^{O(\varepsilon^{-2})} \cdot \mathrm{poly}(\varepsilon^{-1}, \log n)$ query time. Finally, we describe how to modify it to obtain the claimed $O(d^2 + d\,\mathrm{poly}(\varepsilon^{-1}, \log n))$ query time.

2.1 Point of departure: one good block

The algorithm of [Ind00] is stated for the Hamming space; we describe its main idea in Euclidean terms, using the notation of this paper. Suppose that we have linear maps $A_1, \ldots, A_d : \mathbb{R}^d \to \mathbb{R}^b$, which we call *blocks*, with $b = O(\log n/\tau^2)$, such that the blocks together preserve the squared norm of every vector, and each block approximately preserves the rescaled squared distance between every pair of database points; these are properties (2) and (3) below. (In [Ind00], the blocks are obtained by a deterministic "densification" map followed by a partition of the coordinates into blocks, chosen by the method of conditional probabilities.) Since each block is low-dimensional, for every $i \in [d]$ we can afford to build a $(1+\delta)$-approximate nearest neighbor structure $\mathcal{D}_i$ for the projected dataset $A_i P$, where $\delta = \Theta(\varepsilon)$. On a query q, we query every $\mathcal{D}_i$ with $A_i q$, obtaining a point $x_i \in P$, and return the candidate minimizing $\|q - x_i\|_2$.

Fix a query q, let $x^\star$ be a true nearest neighbor, and write $r = \|q - x^\star\|_2$. Since q is not a database point, (3) gives no control over $A_i(q - x^\star)$ for any particular block. However, by (2) and averaging, there is a block i with $\|A_i(q - x^\star)\|_2^2 \le r^2/d$. For this block, the triangle inequality in $\mathbb{R}^b$ and the guarantee of $\mathcal{D}_i$ give $\|A_i(x^\star - x_i)\|_2 \le (2+\delta)\,\|A_i(q - x^\star)\|_2$. Since $x^\star - x_i$ is a difference of database points, (3) transfers this bound back to the original space, and the triangle inequality in $\mathbb{R}^d$ yields

$$\|q - x_i\|_2 \le r + \|x^\star - x_i\|_2 \le r + \sqrt{\tfrac{d}{1-\tau}}\,(2+\delta)\,\|A_i(q - x^\star)\|_2 \le \big(3 + O(\delta + \tau)\big) r.$$

The factor 3 is tight for this argument: in block i, the point $A_i x_i$ may lie on the opposite side of $A_i q$ from $A_i x^\star$, so that $\|x^\star - x_i\|_2 \approx 2r$, while the blocks other than i, on which the argument has no control over q, may align $q - x^\star$ with $x^\star - x_i$.

The loss comes from relying on a single good block. The argument uses the query only through block i, and returns to the original space only through the difference $x^\star - x_i$ of database points, paying for the triangle inequality twice. To avoid this, we would like to use all blocks at once: by (2), the squared distance $\|q - x\|_2^2$ is recovered exactly as $\sum_i \|A_i(q - x)\|_2^2$ for any point x, including directions that involve the query. The obstacle is that the answer of $\mathcal{D}_i$ depends on the block i itself, so the projected distances $\|A_i(q - x_i)\|_2^2$ of the different answers x_i cannot simply be summed to bound the distance of any single candidate. We address this by letting each local structure use

many blocks, so that its answer depends only weakly on any single one of them, and by analyzing the *average* squared distance of the answers through a bias–variance decomposition rather than the triangle inequality. The rest of this section describes this approach.

2.2 Our construction: defining and grouping the blocks

By padding with zeros, we can assume without loss of generality that the dimension d a power of two. We construct d linear maps $A_1, \ldots, A_d : \mathbb{R}^d \to \mathbb{R}^b$ that map vectors in $\mathbb{R}^d$ to smaller "blocks" in $\mathbb{R}^b$ with two properties. First, the blocks preserve the squared norm of every vector:

$$\forall z \in \mathbb{R}^d, \quad \sum_{i=1}^{k} \|A_i z\|_2^2 = \|z\|_2^2 . \tag{2}$$

Second, every difference of database points has approximately the same squared norm in every block:

$$\forall x, y \in P, \quad \forall i \in [d], \quad \frac{1-\tau}{d} \|x-y\|_2^2 \le \|A_i(x-y)\|_2^2 \le \frac{1+\tau}{d} \|x-y\|_2^2 , \tag{3}$$

Here $\tau \in (0,1)$ is an accuracy parameter, which we will set to $\Theta(\varepsilon^2)$.

These linear maps are built by deterministically choosing $b = O(\log n/\tau^2)$ signed Hadamard transforms and grouping together the same output coordinate from each transform. The first property follows from the orthogonality of signed Hadamard transforms. The second property follows from choosing the signs by conditional expectation and a moment bound for sums of squared Rademacher sums. We note that similar Hadamard-based orthogonal block decompositions, with randomly chosen signs, are used in the Las Vegas ANN data structures of Wei [Wei22]; here the signs are chosen deterministically, and concentration is required only for differences of database points.

Once we have these linear maps, we group the blocks together as follows: for an integer $s = O(\varepsilon^{-2})$, we consider every ordered tuple $\omega = (i_1, \ldots, i_s) \in [d]^s$ and concatenate the corresponding blocks:

$$\mathcal{A}_\omega z := A_{i_1} z \oplus \cdots \oplus A_{i_s} z \in \mathbb{R}^{b \cdot s},$$

where $\oplus$ denotes the direct sum operator, i.e., concatenation.

For every ω, we build a deterministic $(1+\Theta(\varepsilon))$-approximate nearest-neighbor structure $\mathcal{D}_\omega$ using the projected dataset $\mathcal{A}_\omega P$. On a query q, each data structure returns a point $x_\omega \in P$. We query every structure and return the candidate minimizing $\|q - x_\omega\|_2$.

2.3 Controlling the dependence on one block

Fix a query q, let $x^\star$ be a true nearest neighbor, and write $r = \|q - x^\star\|_2$. For the analysis, we choose a uniformly random tuple $\Omega = (I_1, \ldots, I_s) \in [d]^s$, then independently choose a uniform position $J \in [s]$, and set $I = I_J$. Thus, I is marginally uniform on $[d]$. These random choices are used only in the proof; the algorithm queries the local structure for every tuple in $[d]^s$.

Let $x_\Omega \in P$ denote the point returned by $\mathcal{D}_\Omega$. Our goal is to bound $\mathbb{E}_\Omega \|q - x_\Omega\|_2^2$. We want to relate it to $\mathbb{E}_\Omega[\|A_i(q - x_\Omega)\|_2^2 \mid I = i]$, the projected expectations, conditioning on $I = i$, such that we could use the approximation guarantee of $\mathcal{D}_\omega$ and the properties of the block maps. Our starting point is to control the difference between two expected locations, before and after conditioning on a block $I = i$:

$$\mathbb{E}_\Omega[x_\Omega \mid I = i] - \mathbb{E}_\Omega[x_\Omega]. \tag{4}$$

We express these expected locations relative to $x^\star$ (which makes it easier to apply Equation (3) latter):

$$a_i = \mathbb{E}_\Omega[x_\Omega - x^\star \mid I = i], \qquad a = \mathbb{E}_\Omega[x_\Omega - x^\star].$$

Therefore, Equation (4) is exactly $a_i - a$. Using the independence of $I_1, \ldots, I_s$, we can show that

$$\mathbb{E}_I \|a_I - a\|_2^2 \le \frac{1}{s}\mathbb{E}_\Omega \|x_\Omega - x^\star - a\|_2^2 . \tag{5}$$

In other words, conditioning on one randomly chosen block $I = i$ changes the expected location of x_Ω by only a small amount. In the next subsection, we use this to obtain a bound on the original distance.

2.4 From conditional, projected distance to the original distance

We are now ready to bound $\mathbb{E}_\Omega \|q - x_\Omega\|_2^2$. By the definition of a, $\mathbb{E}_\Omega[x_\Omega] = x^\star + a$, so we decompose

$$\begin{aligned} \mathbb{E}_\Omega \|q - x_\Omega\|_2^2 &= \|q - \mathbb{E}_\Omega[x_\Omega]\|_2^2 + \mathbb{E}_\Omega \|x_\Omega - \mathbb{E}_\Omega[x_\Omega]\|_2^2 \\ &= \|q - x^\star - a\|_2^2 + \mathbb{E}_\Omega \|x_\Omega - x^\star - a\|_2^2 \end{aligned} \tag{6}$$

This can be viewed as a bias term plus a variance term. We bound these two terms separately, by relating them to the bias and variance terms in the decomposition of $\mathbb{E}_\Omega[\|A_i(q - x_\Omega)\|_2^2 \mid I = i]$. By the first property (Equation (2)) of the block maps,

$$\|q - x^\star - a\|_2^2 = \sum_{i=1}^{d} \|A_i(q - x^\star - a)\|_2^2 ,$$

We can bound the right-hand side via two observations: first, replacing a by a_i introduces only a small additive error. This can be shown using Equation (5) and the fact that although $a_i - a$ need not be a difference of database points, it is a difference of averages of database points, thus one can extend the second property (Equation (3)) of the block maps to apply here. Secondly, once we replace a by a_i, since $\mathbb{E}_\Omega[x_\Omega \mid I = i] = x^\star + a_i$, each summand becomes the bias term in the decomposition of $\mathbb{E}_\Omega[\|A_i(q - x_\Omega)\|_2^2 \mid I = i]$, that is

$$\|A_i(q - \mathbb{E}_\Omega[x_\Omega \mid I = i])\|_2^2 ,$$

which in turn can be expressed as the difference between $\mathbb{E}_\Omega[\|A_i(q - x_\Omega)\|_2^2 \mid I = i]$ and a conditional variance of $A_i x_\Omega$. By Equation (3), this variance is at least

$$\frac{1 - \tau}{d}\mathbb{E}_\Omega[\|x_\Omega - \mathbb{E}_\Omega[x_\Omega \mid I = i]\|_2^2 \mid I = i].$$

Let

$$\mathcal{V} := \mathbb{E}_{\Omega,I} \|x_\Omega - \mathbb{E}_\Omega[x_\Omega \mid I]\|_2^2 = \mathbb{E}_{\Omega,I} \|x_\Omega - x^\star - a_I\|_2^2$$

be the average conditional variance in the original space. Then summing over $i \in [d]$ and using the approximation guarantees of the ANN data structures, we obtain

$$\sum_{i=1}^{d} \|A_i(q - x^\star - a_i)\|_2^2 \le (1 + \Theta(\varepsilon))^2 r^2 - (1 - \tau)\mathcal{V}. \tag{7}$$

Equation (7) not only gives a bound for the bias term of $\mathbb{E}_\Omega \|q - x_\Omega\|_2^2$, but also implies $\mathcal{V} = O(r^2)$. Moving on to the variance term of $\mathbb{E}_\Omega \|q - x_\Omega\|_2^2$, we have that

$$\mathbb{E}_\Omega \|x_\Omega - x^\star - a\|_2^2 = \mathcal{V} + \mathbb{E}_I \|a_I - a\|_2^2 \le \mathcal{V} + \frac{1}{s}\mathbb{E}_\Omega \|x_\Omega - x^\star - a\|_2^2$$

by Equation (5). Thus rearranging gives

$$\mathbb{E}_\Omega \|x_\Omega - x^\star - a\|_2^2 \le \mathcal{V} + \frac{1}{s-1}\mathcal{V}.$$

The $\mathcal{V}$ contribution here cancels the $-(1-\tau)\mathcal{V}$ contribution in Equation (7), up to $\tau\mathcal{V}$. Returning to the decomposition of $\mathbb{E}_\Omega \|q - x_\Omega\|_2^2$ above, we obtain

$$\sqrt{\mathbb{E}_\Omega \|q - x_\Omega\|_2^2} \le \big(1 + \Theta(\varepsilon) + O(s^{-1/2} + \sqrt{\tau})\big) r.$$

Choosing appropriate $s = \Theta(\varepsilon^{-2})$ and $\tau = \Theta(\varepsilon^2)$, makes this bound at most $(1+\varepsilon)r$. Finally, at least one returned point has squared distance no greater than the average, giving the desired approximation.

2.5 Reducing the number of tuples using an expander

The construction above queries all d^s tuples. With $s = \Theta(\varepsilon^{-2})$, this gives the factor $d^{O(\varepsilon^{-2})}$ in the query time. We now explain how to reduce the number of local structures to d. The resulting construction gives the improved query bound in Theorem 1.1.

The crucial observation is that the analysis does not require all tuples. Apart from the conditional-mean estimate, it only uses the fact that a uniformly chosen block occurrence is uniform on $[d]$. Thus it suffices to find a smaller family with uniform block frequencies and a similar bound on the dependence between an answer and one of its constituent blocks.

For this purpose, let G be a constant-degree regular undirected expander on the block indices. For each vertex v, form a tuple containing the endpoints of all walks of length ℓ starting at v, retaining repeated endpoints. If the degree is Δ, every tuple has Δ^ℓ entries, and there are only d tuples. We build one local ANN structure for each of these tuples. Let $x_v \in P$ denote the answer returned by the structure associated with starting vertex v. We again return the candidate closest to the query in the original space.

To see why this family has the required properties, choose a starting vertex V uniformly and choose one of its walks uniformly. Denote the endpoint by I. Regularity makes I uniform. Moreover, if W is the normalized adjacency matrix of G, conditioning on $I = i$ averages the answers according to the i-th row of W^ℓ. If the nontrivial eigenvalues of W have absolute value at most $\theta < 1$, this averaging contracts mean-zero functions by at most $\rho = \theta^\ell$. Define $a_i = \mathbb{E}[x_V - x^\star \mid I = i]$ and $a = \mathbb{E}[x_V - x^\star]$ as before. The conditional means then satisfy

$$\mathbb{E} \|a_I - a\|_2^2 \le \frac{\rho^2}{1-\rho^2}\mathbb{E} \|x_V - x^\star - a_I\|_2^2 .$$

Here, the factor $\frac{\rho^2}{1-\rho^2}$ replaces the factor $1/(s-1)$ in the exhaustive construction. The remaining argument gives approximation $1 + \delta + O(\rho + \sqrt{\tau})$.

Taking a suitable $\ell = O(\log(1/\varepsilon))$ makes $\rho = O(\varepsilon)$, while each tuple still contains only $\Delta^\ell = (1/\varepsilon)^{O(1)}$ blocks. The local dimension is therefore polynomial in $1/\varepsilon$ and logarithmic in n. There are now only $O(d)$ local queries. Their total cost, together with evaluation of the block maps, is $d\,\mathrm{poly}(\varepsilon^{-1}, \log nd)$; examining d returned candidates in the original space $\mathbb{R}^d$ costs $O(d^2)$.

We note that eplacing an exhaustive family of tuples by tuples generated by walks on an expander is a classical derandomization technique, used, e.g., for graph products [AFWZ95]. Closest to our setting, [KKKÓC20] use expanders to derandomize the tensoring-based correlation amplification behind Valiant's algorithm for finding outlier correlations. Similarly, [ACW20] used expander walks to derandomize the polynomial method for offline approximate nearest neighbor search.

3 Preliminaries

Notations. For an integer $m \geq 1$, let $[m] = \{1, \ldots, m\}$. All norms are Euclidean. We use $u \oplus v$ to denote concatenation. Throughout the paper, $P \subseteq \mathbb{R}^d$ is a nonempty set of $n \geq d$ points, $d \geq 1$ is a power of two, and $\varepsilon \in (0, 1)$.

Definition 3.1 (Balanced blocks.)**.** Let $A_1, \ldots, A_d : \mathbb{R}^d \to \mathbb{R}^b$ be linear maps, and let $\tau \in (0, 1)$. We say that they form a τ-balanced block system for P if

$$\sum_{i=1}^{d} \|A_i z\|_2^2 = \|z\|_2^2 \qquad \text{for every } z \in \mathbb{R}^d, \tag{8}$$

and, for every $x, y \in P$ and every $i \in [d]$,

$$\frac{1-\tau}{d} \|x - y\|_2^2 \leq \|A_i(x - y)\|_2^2 \leq \frac{1+\tau}{d} \|x - y\|_2^2 . \tag{9}$$

The first property is also called the Parseval identity. Notice that (8) holds for every vector, whereas (9) is required only for differences between database points.

We state the existence of such a balanced blokc system as follows:

Theorem 3.2. *Given $P \subseteq \mathbb{R}^d$ and $\tau \in (0, 1)$, one can deterministically construct a τ-balanced block system with*

$$b = O\big(\tau^{-2} \log n\big).$$

The construction takes $\mathrm{poly}(n, d, 1/\tau)$ arithmetic operations. After preprocessing, all d maps can be evaluated on a vector in $O(bd \log n)$ time.

We prove Theorem 3.2 in Section 6.

We also need the following consequence of the approximate Voronoi diagram construction of Har-Peled [HP01] and its treatment in Har-Peled, Indyk, and Motwani [HPIM12]. We state a polynomial preprocessing bound and a conservative query bound; these suffice for our result.

Lemma 3.3. *Let $Y \subseteq \mathbb{R}^m$ be nonempty and contain at most n points, where $m \geq 1$, and let $\delta \in (0, 1/2)$. There is a deterministic data structure that returns a $(1+\delta)$-approximate nearest neighbor for every query in $\mathbb{R}^m$. For an absolute constant $C > 1$, its space is*

$$n(C/\delta)^{O(m)} \operatorname{polylog}(n/\delta),$$

its preprocessing time is

$$\mathrm{poly}(n, m)(C/\delta)^{O(m)} \operatorname{polylog}(n/\delta),$$

and its query time is $O(m^2 \log^2(n/\delta))$.

For completeness, we include a deterministic implementation sufficient for these bounds in Appendix A.

We will apply this lemma in dimension logarithmic in n, up to a factor polynomial in $1/\varepsilon$. The dependence $(C/\delta)^{O(m)}$ therefore gives space polynomial in n for fixed ε. The other polynomial factors in preprocessing and query time will not affect the final bounds.

Finally, for a square-integrable random vector X and an independent copy X', we will use some basic identities

Lemma 3.4.

$$\mathbb{E}\,\|X-\mathbb{E}X\|_2^2=\frac{1}{2}\mathbb{E}\,\|X-X'\|_2^2\,,\qquad \mathbb{E}\,\|v-X\|_2^2=\|v-\mathbb{E}X\|_2^2+\mathbb{E}\,\|X-\mathbb{E}X\|_2^2\,. \tag{10}$$

Both follow by expanding the squared norms.

4 The exhaustive construction

In this section, we first describe our exhaustic construction that uses all ordered tuples of blocks. Then we state the correctness guarantee and the time and space bounds. The proof of the correctness appear in the next Section 5.

4.1 Preprocessing and query algorithm

Preprocessing. Fix parameters $\delta,\tau\in(0,1/2)$ and an integer $s\geq 2$, to be chosen below, and let $\alpha=q+\delta$. Construct the maps $A_1,\dots,A_d$ using Theorem 3.2. For each ordered tuple $\omega=(i_1,\dots,i_s)\in[d]^s$, define

$$\mathcal{A}_\omega z=A_{i_1}z\oplus\cdots\oplus A_{i_s}z\in\mathbb{R}^{sb}.$$

We refer to this concatenation as a superblock. Repeated indices are allowed, and the corresponding blocks are retained with their multiplicities. Build the data structure of Lemma 3.3 on the projected database $\mathcal{A}_\omega P$ with error parameter δ, keeping a pointer from each projected point to its original point in P.

Answering a query. Given $q\in\mathbb{R}^d$, first compute $A_1q,\dots,A_dq$. Then query each local data structure with $\mathcal{A}_\omega q$, obtaining a point $x_\omega\in P$ such that

$$\|\mathcal{A}_\omega(q-x_\omega)\|_2\leq\alpha\min_{x\in P}\|\mathcal{A}_\omega(q-x)\|_2\,. \tag{11}$$

Compute $\|q-x_\omega\|_2$ for every returned point, and return a candidate minimizing this distance. All ties are resolved according to a fixed ordering of P. The preprocessing and query procedures are deterministic.

4.2 Corretness guarantee

The following theorem shows that one of the returned candidates is close to q in the original space, and thus so does the query algorithm.

Theorem 4.1. *Let $A_1,\dots,A_d$ be a τ-balanced block system for P, with $0<\tau<1$, and let $s\geq 2$. Suppose that for every $\omega\in[d]^s$, a point $x_\omega\in P$ satisfies (11) with factor $\alpha\geq 1$. Then, for every query q,*

$$\min_{\omega\in[d]^s}\|q-x_\omega\|_2\leq C_{\alpha,\tau,s}\min_{x\in P}\|q-x\|_2\,,$$

where

$$C_{\alpha,\tau,s} = \frac{\alpha + \sqrt{2\alpha^2/(s-1) + 12\tau(1+\alpha)^2}}{\sqrt{1-\tau}}.$$

In particular, if α is bounded and $\tau \le 1/2$, then $C_{\alpha,\tau,s} = \alpha + O(s^{-1/2} + \sqrt{\tau})$.

We prove this theorem in Section 5.

4.3 Parameters and time and space bounds

For a target error $\varepsilon \in (0,1)$, set

$$\delta = \frac{\varepsilon}{100}, \qquad \tau = \frac{\varepsilon^2}{10^6}, \qquad s = 1 + \left\lceil \frac{10^4}{\varepsilon^2} \right\rceil. \tag{12}$$

Substituting these choices and $\alpha = 1 + \delta$ in Theorem 4.1 gives approximation factor at most $1 + \varepsilon$. Each local structure has dimension $sb = O(\varepsilon^{-6} \log n)$, and there are $d^s = d^{O(\varepsilon^{-2})}$ tuples. Since $n \ge d$, the block construction and Lemma 3.3 give preprocessing time and space $n^{\mathrm{poly}(1/\varepsilon)}$, and query time $d^{O(\varepsilon^{-2})}\,\mathrm{poly}(\varepsilon^{-1}, \log n)$, including the evaluation of the block maps and the distance checks for all returned candidates. Section 7 improves the query time by reducing the number of local structures to $O(d)$.

5 Proof of Theorem 4.1

In this section, we prove Theorem 4.1. Fix a query q, let $x^\star \in P$ be a nearest neighbor of q, and write $r = \|q - x^\star\|_2$. We will show that $\mathbb{E}_\Omega \|q - x_\Omega\|_2^2 \le C_{\alpha,\tau,s}^2 r^2$, where $\Omega = (I_1, \ldots, I_s)$ is uniform in $[d]^s$. Since the algorithm returns the closest candidate in the original space, this implies the claimed approximation guarantee.

As described in the technical overview, we independently choose J uniformly in $[s]$ and set $I = I_J$. Thus I is uniform in $[d]$, but it can be correlated with the returned point x_Ω. These random choices are used only for the analysis. All expectations below are over this two-step selections of Ω and I unless another distribution is specified.

5.1 Controlling the dependence on one block

In order to bound $\mathbb{E}_\Omega \|q - x_\Omega\|_2^2$, we want to relate it to the projected expectations, conditioning on $I = i$, such that we could use the approximation guarantee of $\mathcal{D}_\omega$ and the properties of the balanced block system. Our starting point is to control the difference between two expected locations, before and after conditioning on a block $I = i$:

$$\mathbb{E}_\Omega[x_\Omega \mid I = i] - \mathbb{E}_\Omega[x_\Omega]. \tag{13}$$

For convenience, we express these expected locations relative to $x^\star$. This choice will be useful when we later apply block balance to differences $x - x^\star$ between database points. Let

$$a_i = \mathbb{E}_\Omega[x_\Omega - x^\star \mid I = i], \qquad a = \mathbb{E}_\Omega[x_\Omega - x^\star] = \frac{1}{d}\sum_{i=1}^{d} a_i.$$

By these definitions, the difference in Equation (13) is exactly $a_i - a$. We next bound its average squared distance over I, namely $\mathbb{E}_I \|a_I - a\|_2^2$.

Lemma 5.1. *We have*

$$\mathbb{E}_I \|a_I - a\|_2^2 \le \frac{1}{s}\mathbb{E}_\Omega \|x_\Omega - x^\star - a\|_2^2 .$$

Proof. Conditioning on $I = i$, we have $\mathbb{E}_\Omega[x_\Omega - x^\star - a \mid I = i] = a_i - a$. Since $a_I - a$ is fixed under this conditioning, averaging over I gives the first equality below.

$$\begin{aligned}\mathbb{E}_I \|a_I - a\|_2^2 &= \mathbb{E}_{\Omega,I} \langle x_\Omega - x^\star - a, a_I - a\rangle \\ &= \mathbb{E}_\Omega \left\langle x_\Omega - x^\star - a, \frac{1}{s}\sum_{j=1}^{s}(a_{I_j} - a)\right\rangle .\end{aligned}$$

The second equality holds by conditioning instead on Ω and average over the independent uniform choice of $J \in [s]$ for $I = I_J$. Each I_j is uniform on $[d]$, so $\mathbb{E}_\Omega[a_{I_j} - a] = 0$. Since $I_1, \dots, I_s$ are independent, the cross terms vanish, giving

$$\mathbb{E}_\Omega \left\| \frac{1}{s}\sum_{j=1}^{s}(a_{I_j} - a)\right\|^2 = \frac{1}{s^2}\sum_{j=1}^{s}\mathbb{E}_\Omega \left\|a_{I_j} - a\right\|_2^2 = \frac{1}{s}\mathbb{E}_I \|a_I - a\|_2^2 .$$

Consequently, Cauchy–Schwarz implies

$$\begin{aligned}\left(\mathbb{E}_I \|a_I - a\|_2^2\right)^2 &\le \mathbb{E}_\Omega \|x_\Omega - x^\star - a\|_2^2 \cdot \mathbb{E}_\Omega \left\| \frac{1}{s}\sum_{j=1}^{s}(a_{I_j} - a)\right\|^2 \\ &= \frac{1}{s}\,\mathbb{E}_\Omega \|x_\Omega - x^\star - a\|_2^2 \cdot \mathbb{E}_I \|a_I - a\|_2^2 .\end{aligned}$$

If $\mathbb{E}_I \|a_I - a\|_2^2 = 0$, the claim is immediate. Otherwise, dividing by this quantity proves the lemma. □

In English, Lemma 5.1 controls the dependence of x_Ω on an observed block I, in the sense that revealing I changes the expected location of x_Ω by only a small amount.

5.2 Decomposing the original distance

By our definition,

$$x^\star + a_i = \mathbb{E}_\Omega[x_\Omega \mid I = i], \qquad x^\star + a_I = \mathbb{E}_\Omega[x_\Omega \mid I], \qquad x^\star + a = \mathbb{E}_\Omega[x_\Omega].$$

We now bound the original distance $\mathbb{E}_\Omega \|q - x_\Omega\|_2^2$. We start by decomposing

$$\begin{aligned}\mathbb{E}_\Omega \|q - x_\Omega\|_2^2 &= \|q - \mathbb{E}_\Omega[x_\Omega]\|_2^2 + \mathbb{E}_\Omega \|x_\Omega - \mathbb{E}_\Omega[x_\Omega]\|_2^2 \\ &= \|q - x^\star - a\|_2^2 + \mathbb{E}_\Omega \|x_\Omega - x^\star - a\|_2^2\end{aligned} \tag{14}$$

Let

$$\mathcal{V} := \mathbb{E}_{\Omega,I} \|x_\Omega - x^\star - a_I\|_2^2 .$$

Then

$$\mathbb{E}_\Omega \|x_\Omega - x^\star - a\|_2^2 = \mathcal{V} + \mathbb{E}_I \|a_I - a\|_2^2 .$$

Substituting this into Lemma 5.1 and rearranging, we obtain

$$\mathbb{E}_I \|a_I - a\|_2^2 \le \frac{\mathcal{V}}{s-1}.$$

thus Equation (14) implies

$$\mathbb{E}_\Omega \|q - x_\Omega\|_2^2 \le \|q - x^\star - a\|_2^2 + \mathcal{V} + \frac{\mathcal{V}}{s-1}. \tag{15}$$

To bound the first term, we use the first property of the balanced block system (Equation (8)):

$$\|q - x^\star - a\|_2^2 = \sum_{i=1}^{d} \|A_i(q - x^\star - a)\|_2^2 . \tag{16}$$

We observe that the right-hand side is quite similar to

$$\sum_{i=1}^{d} T_i, \quad T_i = \|A_i(q - x^\star - a_i)\|_2^2 ,$$

which can be easily bounded using the approximation guarantee of $\mathcal{D}_\omega$. Therefore, we first bound $\sum T_i$. Then in the next subsection, we will bound the error caused by replacing a_i by a.

Lemma 5.2. *The approximation guarantees of $\mathcal{D}_\omega$ imply*

$$\sum_{i=1}^{d} T_i + (1-\tau)\mathcal{V} \le \alpha^2 r^2. \tag{17}$$

In particular,

$$\mathcal{V} \le \frac{\alpha^2 r^2}{1-\tau}, \qquad \sum_{i=1}^{d} T_i + \mathcal{V} \le \frac{\alpha^2 r^2}{1-\tau}. \tag{18}$$

Proof. Since $x^\star$ is a candidate in the dataset, the approximation guarantees of $\mathcal{D}_\omega$ (Equation (11)) give

$$\|\mathcal{A}_\Omega(q - x_\Omega)\|_2^2 \le \alpha^2 \|\mathcal{A}_\Omega(q - x^\star)\|_2^2 .$$

Averaging over the uniformly chosen position and then over Ω gives

$$\mathbb{E}_\Omega \|\mathcal{A}_\Omega(q - x_\Omega)\|_2^2 = s\, \mathbb{E}_{\Omega,I} \|A_I(q - x_\Omega)\|_2^2 .$$

On the other side, uniformity of I and (8) give $\mathbb{E}_\Omega \|\mathcal{A}_\Omega(q - x^\star)\|_2^2 = sr^2/d$. Thus

$$\frac{d}{s}\, \mathbb{E}_\Omega \|\mathcal{A}_\Omega(q - x_\Omega)\|_2^2 = d\, \mathbb{E}_{\Omega,I} \|A_I(q - x_\Omega)\|_2^2 \le \alpha^2 r^2. \tag{19}$$

Fix i and condition on $I = i$. The conditional mean of x_Ω is $x^\star + a_i$, so

$$\mathbb{E}_\Omega[\|A_i(q - x_\Omega)\|_2^2 \mid I = i] = T_i + \mathbb{E}_\Omega[\|A_i(x_\Omega - x^\star - a_i)\|_2^2 \mid I = i]. \tag{20}$$

To bound the second term from below, let X, X' be independent points with the conditional distribution of x_Ω given $I = i$. Both lie in P, so the variance identity and block balance imply

$$\begin{aligned}
\mathbb{E}_\Omega[\|A_i(x_\Omega - x^\star - a_i)\|_2^2 \mid I = i] &= \frac{1}{2}\mathbb{E}\left\|A_i(X - X')\right\|_2^2 \\
&\ge \frac{1-\tau}{2d}\mathbb{E}\left\|X - X'\right\|_2^2 \\
&= \frac{1-\tau}{d}\mathbb{E}_\Omega[\|x_\Omega - x^\star - a_i\|_2^2 \mid I = i].
\end{aligned}$$

Summing (20) over i now gives

$$d\,\mathbb{E}_{\Omega,I}\,\|A_I(q-x_\Omega)\|_2^2 \geq \sum_{i=1}^{d} T_i + (1-\tau)\mathcal{V}.$$

Combining this with (19) proves (17). Its first consequence follows by dropping $\sum_i T_i$. Adding $\tau\mathcal{V}$ to (17) and using this consequence proves the second. □

5.3 Justifying replacing a_i by a

To use Lemma 5.2 in (16), we need to control $\sum_i \|A_i(a_i-a)\|_2^2$. Lemma 5.1 bounds the differences $a_i - a$ in the original space. These vectors need not be differences between database points, but they are differences of averages of database points. We use this observation to extend the balance guarantee.

Lemma 5.3. *We have*

$$\sum_{i=1}^{d} \|A_i(a_i-a)\|_2^2 \leq \mathbb{E}_I\,\|a_I-a\|_2^2 + 12\tau\,\mathbb{E}_\Omega\,\|x_\Omega - x^\star\|_2^2\,.$$

Proof. Fix i. For any $x, y \in P$, the three vectors $x-x^\star$, $y-x^\star$, and $x-y$ are differences between database points. Polarization and (9) therefore give

$$\begin{aligned}
&|d\,\langle A_i(x-x^\star), A_i(y-x^\star)\rangle - \langle x-x^\star, y-x^\star\rangle| \\
&\quad\leq \frac{\tau}{2}\big(\|x-x^\star\|_2^2 + \|y-x^\star\|_2^2 + \|x-y\|_2^2\big) \\
&\quad\leq \frac{3\tau}{2}\big(\|x-x^\star\|_2^2 + \|y-x^\star\|_2^2\big).
\end{aligned}$$

Let μ_i be the conditional distribution of x_Ω given $I=i$, and let μ be its unconditional distribution. Write $c_x = \mu_i(x) - \mu(x)$. Then

$$a_i - a = \sum_{x\in P} c_x(x-x^\star), \qquad \sum_{x\in P}|c_x| \leq 2.$$

Expanding the squared norms and applying the preceding inner-product bound, we obtain

$$\begin{aligned}
&\Big|d\,\|A_i(a_i-a)\|_2^2 - \|a_i-a\|_2^2\Big| \\
&\quad\leq \frac{3\tau}{2}\sum_{x,y\in P}|c_x||c_y|\big(\|x-x^\star\|_2^2 + \|y-x^\star\|_2^2\big) \\
&\quad= 3\tau\left(\sum_{y\in P}|c_y|\right)\left(\sum_{x\in P}|c_x|\,\|x-x^\star\|_2^2\right) \\
&\quad\leq 6\tau\left(\mathbb{E}_\Omega[\|x_\Omega-x^\star\|_2^2 \mid I=i] + \mathbb{E}_\Omega\,\|x_\Omega-x^\star\|_2^2\right).
\end{aligned}$$

The last step uses $|c_x| \leq \mu_i(x) + \mu(x)$. Summing over i and dividing by d proves the lemma. □

The additive error above depends on the distance from a returned point to $x^\star$. We next show that its second moment is bounded by a constant multiple of r^2 when α is bounded and $\tau \leq 1/2$.

Lemma 5.4. *We have*

$$\mathbb{E}_\Omega \|x_\Omega - x^\star\|_2^2 \le \frac{(1+\alpha)^2}{1-\tau} r^2.$$

Proof. For every tuple ω, the triangle inequality and (11) give

$$\begin{aligned}\|\mathcal{A}_\omega(x_\omega - x^\star)\|_2 &\le \|\mathcal{A}_\omega(q - x_\omega)\|_2 + \|\mathcal{A}_\omega(q - x^\star)\|_2 \\ &\le (1+\alpha)\,\|\mathcal{A}_\omega(q - x^\star)\|_2 .\end{aligned}$$

Since $x_\omega, x^\star \in P$, block balance applies to each of the s block occurrences in ω:

$$\|\mathcal{A}_\omega(x_\omega - x^\star)\|_2^2 \ge \frac{s(1-\tau)}{d}\,\|x_\omega - x^\star\|_2^2 .$$

Squaring the first bound and averaging over Ω therefore yields

$$\frac{s(1-\tau)}{d}\mathbb{E}_\Omega \|x_\Omega - x^\star\|_2^2 \le (1+\alpha)^2 \mathbb{E}_\Omega \|\mathcal{A}_\Omega(q - x^\star)\|_2^2 = \frac{s(1+\alpha)^2}{d} r^2.$$

The equality uses (8) and uniform block marginals. Canceling s/d proves the claim. □

Combining lemmas 5.1, 5.3 and 5.4 gives the desired replacement bound:

$$\sum_{i=1}^{d} \|A_i(a_i - a)\|_2^2 \le \frac{\mathcal{V}}{s-1} + \frac{12\tau(1+\alpha)^2}{1-\tau} r^2. \tag{21}$$

5.4 Completing the proof

We now combine the original-distance decomposition with the bounds on the projected distances and the error from replacing a_i by a.

Proof of Theorem 4.1. By Equations (15) and (16),

$$\mathbb{E}_\Omega \|q - x_\Omega\|_2^2 \le \sum_{i=1}^{d} \|A_i(q - x^\star - a)\|_2^2 + \mathcal{V} + \frac{\mathcal{V}}{s-1}.$$

For each i, write

$$A_i(q - x^\star - a) = A_i(q - x^\star - a_i) + A_i(a_i - a).$$

Applying the triangle inequality to the concatenation of these block vectors and two scalar coordinates gives

$$\sqrt{\mathbb{E}_\Omega \|q - x_\Omega\|_2^2} \le \sqrt{\sum_{i=1}^{d} T_i + \mathcal{V}} + \sqrt{\sum_{i=1}^{d} \|A_i(a_i - a)\|_2^2 + \frac{\mathcal{V}}{s-1}}. \tag{22}$$

Here we place $\sqrt{\mathcal{V}}$ in the first vector and $\sqrt{\mathcal{V}/(s-1)}$ in a separate coordinate of the second vector.

The first square root is at most $\alpha r/\sqrt{1-\tau}$ by Equation (18). For the second, the replacement bound Equation (21) and the bound on $\mathcal{V}$ in Equation (18) give

$$\begin{aligned}\sum_{i=1}^{d} \|A_i(a_i - a)\|_2^2 + \frac{\mathcal{V}}{s-1} &\le \frac{2\mathcal{V}}{s-1} + \frac{12\tau(1+\alpha)^2}{1-\tau} r^2 \\ &\le \frac{r^2}{1-\tau}\left(\frac{2\alpha^2}{s-1} + 12\tau(1+\alpha)^2\right).\end{aligned}$$

Substituting these bounds into Equation (22), we obtain

$$\sqrt{\mathbb{E}_\Omega \|q - x_\Omega\|_2^2} \le \frac{\alpha + \sqrt{2\alpha^2/(s-1) + 12\tau(1+\alpha)^2}}{\sqrt{1-\tau}}\, r = C_{\alpha,\tau,s} r.$$

Finally, at least one tuple has squared distance no greater than this average. Its returned point therefore satisfies $\|q - x_\omega\|_2 \le C_{\alpha,\tau,s} r$, proving the theorem. □

6 Deterministic construction of the balanced blocks

In this section, we prove Theorem 3.2. We first describe a family of maps for which the Parseval identity (Equation (8)) holds for every choice of signs. We then choose the signs by conditional expectation so that every database difference is balanced (Equation (9)) in every block.

6.1 The signed Hadamard construction

Let $H \in \mathbb{R}^{d\times d}$ be the normalized Walsh–Hadamard matrix, with entries $H_{jr} = d^{-1/2} h_{jr}$ for $h_{jr} \in \{-1, +1\}$. Thus $H^\mathsf{T} H = I_d$, and multiplication by H takes $O(d \log n)$ time.

For an integer L to be chosen below, take sign vectors $\xi^{(1)}, \dots, \xi^{(L)} \in \{-1, +1\}^d$. For each $j \in [d]$, define $B_j : \mathbb{R}^d \to \mathbb{R}^L$ by

$$(B_j z)_\ell = \sqrt{d}\,(H \operatorname{diag}(\xi^{(\ell)}) z)_j = \sum_{r=1}^{d} h_{jr} \xi_r^{(\ell)} z_r, \qquad A_j = (Ld)^{-1/2} B_j.$$

In other words, block j collects the jth coordinate from each of L signed Hadamard transforms. For any assignment of the signs,

$$\sum_{j=1}^{d} \|A_j z\|_2^2 = \frac{1}{L} \sum_{\ell=1}^{L} \left\| H \operatorname{diag}(\xi^{(\ell)}) z \right\|_2^2 = \|z\|_2^2 .$$

Thus Parseval holds exactly. Moreover, all blocks can be evaluated together using L signed Hadamard transforms, in $O(Ld \log n)$ time. It remains to choose the signs so that block balance holds with $L = O(\tau^{-2} \log n)$.

6.2 A moment bound

We use the following estimate to construct a potential whose expectation is small under random signs.

Lemma 6.1. *Let $Y = \sum_r a_r \sigma_r$, where the σ_r are independent uniform signs. Write $v = \sum_r a_r^2$ and $X = Y^2 - v$. If $X_1, \dots, X_L$ are independent copies of X, then for every integer $1 \le p \le L$,*

$$\mathbb{E} \left| \sum_{\ell=1}^{L} X_\ell \right|^{2p} \le (CpL)^p v^{2p},$$

where C is an absolute constant.

Proof. The claim is immediate for $v = 0$; otherwise rescale so that $v = 1$. The inequality $\cosh t \le e^{t^2/2}$ gives $\mathbb{E}e^{tY} \le e^{t^2/2}$ and hence $(\mathbb{E}|Y|^{2q})^{1/(2q)} \le C_1\sqrt{q}$ for every integer $q \ge 1$. It follows that $\mathbb{E}|X|^q \le (C_2 q)^q$. Since $\mathbb{E}X = 0$, expanding the exponential and summing the terms of order at least two yields

$$\mathbb{E}e^{\lambda X} \le e^{C_3\lambda^2} \qquad (|\lambda| \le c_0).$$

Independence and the Chernoff bound then give

$$\mathbb{P}\left[\left|\sum_{\ell=1}^{L} X_\ell\right| \ge t\right] \le 2\exp\bigl(-c_1 \min\bigl\{t^2/L, t\bigr\}\bigr).$$

Integrating this tail against $2p\,t^{2p-1}$ bounds the $2p$th moment by $C_4^{2p}((pL)^p + p^{2p})$. Since $p \le L$, this is at most $(CpL)^p$ after adjusting C. Rescaling proves the lemma. □

6.3 Choosing the signs by conditional expectation

Let $\mathcal{D}$ contain one orientation of each nonzero difference between two database points, and let $N = |\mathcal{D}| \le n(n-1)/2$. If $N = 0$, choose arbitrary signs and any $L = \Theta(\tau^{-2}\log n)$; the balance requirement is vacuous. Hence assume $N \ge 1$.

For $v \in \mathcal{D}$, put $r_v = \|v\|_2^2$, and define

$$Y_{\ell,j,v} = (B_j v)_\ell, \qquad X_{\ell,j,v} = Y_{\ell,j,v}^2 - r_v.$$

For a fixed pair (j, v), the variables $X_{\ell,j,v}$ are independent over ℓ when all signs are independent and uniform. Also,

$$d\,\|A_j v\|_2^2 - r_v = \frac{1}{L}\sum_{\ell=1}^{L} X_{\ell,j,v}. \tag{23}$$

Choose

$$p_0 = \lceil \log_4(8Nd) \rceil, \qquad L = \lceil C_0 \tau^{-2} p_0 \rceil,$$

for a sufficiently large absolute constant C_0, and consider the nonnegative potential

$$\Phi = \sum_{v\in\mathcal{D}} \sum_{j=1}^{d} \left(\frac{d\,\|A_j v\|_2^2 - r_v}{\tau r_v} \right)^{2p_0}. \tag{24}$$

If $\Phi < 1$, every summand is smaller than one, which is exactly the desired balance guarantee. We next show that a suitable assignment can be found deterministically.

By Lemma 6.1 and (23),

$$\mathbb{E}\Phi \le Nd\left(\frac{Cp_0}{L\tau^2}\right)^{p_0} \le Nd\,4^{-p_0} \le \frac{1}{8},$$

where the second inequality follows from the choice of C_0. To derandomize this estimate, we need to compute the expectation after an arbitrary subset of the signs has been fixed.

Lemma 6.2. *For any partial assignment of the signs, the conditional expectation of Φ can be computed using $\mathrm{poly}(N, d, L, p_0)$ arithmetic operations.*

Proof. Fix (ℓ, j, v). Under the partial assignment, $Y_{\ell,j,v} = c + \sum_{r \in R} a_r \sigma_r$, where c, a_r are known and the remaining signs σ_r are independent. The polynomial

$$e^{ct} \prod_{r \in R} \cosh(a_r t) \pmod{t^{4p_0+1}}$$

is the truncated exponential generating function of its conditional moments. Multiplying its coefficient of t^a by $a!$ recovers the ath moment for $a \le 4p_0$. By the binomial formula, these moments determine the conditional moments of $X_{\ell,j,v} = Y_{\ell,j,v}^2 - r_v$ through order $2p_0$.

The remaining randomness in different repetitions is independent. Multiply the L exponential generating functions of $X_{\ell,j,v}$, truncating at degree $2p_0$. This recovers the conditional $2p_0$th moment of their sum. Dividing by $(L\tau r_v)^{2p_0}$ and summing over (j, v) gives the conditional expectation of (24). All polynomial products have degree $O(p_0)$, so even naive convolution uses polynomially many arithmetic operations. □

Proof of Theorem 3.2. Expose the Ld signs in a fixed order. At each step, use Lemma 6.2 to compute the conditional expectation of Φ for each of the two possible values of the next sign, and choose one with the smaller value. Their average is the current conditional expectation, so this choice never increases it. Once all signs are fixed,

$$\Phi \le \mathbb{E}\Phi \le \frac{1}{8} < 1.$$

Consequently, for every $v \in \mathcal{D}$ and $j \in [d]$,

$$\left| d \left\| A_j v \right\|_2^2 - \left\| v \right\|_2^2 \right| < \tau \left\| v \right\|_2^2 .$$

This proves block balance with $b = L$. The zero differences satisfy it automatically, and Parseval holds for every sign assignment. Since $p_0 = O(\log n)$ and $L = O(\tau^{-2} \log n)$, the number of arithmetic operations is polynomial in $n, d, 1/\tau$. The simultaneous evaluation bound follows from the signed Hadamard construction. □

For rational input and rational τ, the potential and its conditional expectations can be evaluated using exact rational arithmetic: the normalization in A_j cancels in (23). This observation concerns the sign-selection procedure; we do not claim a bit-complexity bound for the full data structure.

7 Reducing the number of tuples using an expander

We now replace the exhaustive family $[d]^s$ by d tuples. As before, we build a local ANN structure for every tuple and return the closest reported point in the original space. To analyze this smaller family, we only need to replace the conditional-mean estimate from Lemma 5.1. All other estimates in Section 5 use the local approximation guarantees, the block properties, and the uniform marginal distribution of the selected block occurrence.

7.1 The tuple family

We use an explicit constant-degree spectral expande, stated in the following Lemma 7.1. This is given by the Margulis–Gabber–Galil construction; we described the construction in more details in Section B.

Lemma 7.1. *There are absolute constants $\Delta \geq 2$ and $\theta_0 \in (0,1)$ such that, for every power of two d, one can deterministically construct a Δ-regular undirected multigraph on $[d]$ whose normalized adjacency matrix W satisfies*

$$\|Wf\|_2 \leq \theta_0 \|f\|_2 \qquad \text{for every } f \in \mathbb{R}^d \text{ with } \sum_{v=1}^{d} f(v) = 0.$$

The neighbor lists can be constructed in $O(d)$ arithmetic operations. Loops and parallel edges are allowed, and each neighbor occurrence is counted in the degree.

Let G be the graph from Lemma 7.1, with normalized adjacency matrix W. The lemma applies since d is a power of two. Fix a positive integer ℓ, and, for each vertex v, let S_v be the tuple of endpoints of all length-ℓ walks starting at v. We use a fixed ordering of these walks and retain every repeated endpoint. Each tuple therefore has length $t = \Delta^\ell$.

For each $v \in [d]$, define

$$\mathcal{A}_v z = \bigoplus_{i \in S_v} A_i z \in \mathbb{R}^{tb},$$

where every occurrence contributes a block. Build the local ANN structure of Lemma 3.3 using the projected dataset $\{\mathcal{A}_v x : x \in P\}$. On a query q, compute all original block vectors and query these d structures. Denote their answers by $x_v \in P$, so that

$$\|\mathcal{A}_v(q - x_v)\|_2 \leq \alpha \min_{x \in P} \|\mathcal{A}_v(q - x)\|_2, \qquad \alpha = 1 + \delta. \tag{25}$$

Return the point x_v minimizing $\|q - x_v\|_2$.

7.2 Controlling the dependence on one block

Fix a query q and a true nearest neighbor $x^\star$, and write $r = \|q - x^\star\|_2$. Choose V uniformly from $[d]$ and then choose a uniformly random occurrence I from S_V. Reversing walks shows that every block occurs exactly t times in the collection of tuples. Therefore I is uniform on $[d]$, and

$$\mathbb{P}[V = v \mid I = i] = (W^\ell)_{iv}.$$

As in the exhaustive construction, let

$$a_i = \mathbb{E}_V[x_V - x^\star \mid I = i], \qquad a = \mathbb{E}_V[x_V - x^\star], \qquad \mathcal{V} = \mathbb{E}_{V,I} \|x_V - x^\star - a_I\|_2^2.$$

The following lemma gives the required replacement for Lemma 5.1.

Lemma 7.2. *For $\rho = \theta_0^\ell < 1$, we have*

$$\mathbb{E}_I \|a_I - a\|_2^2 \leq \frac{\rho^2}{1 - \rho^2} \mathcal{V}.$$

Proof. Since $\sum_v (x_v - x^\star - a) = 0$, the spectral bound for W^ℓ applies to each coordinate of these vectors. Using the conditional distribution above gives

$$\begin{aligned} \mathbb{E}_I \|a_I - a\|_2^2 &= \frac{1}{d} \sum_i \left\| \sum_v (W^\ell)_{iv} (x_v - x^\star - a) \right\|^2 \\ &\leq \rho^2 \mathbb{E}_V \|x_V - x^\star - a\|_2^2 \\ &= \rho^2 \big(\mathcal{V} + \mathbb{E}_I \|a_I - a\|_2^2 \big), \end{aligned}$$

where the last equality is the variance decomposition. Rearranging proves the claim. □

7.3 Bounding the original distance

We explain how the proof from Section 5 applies to this family. Averaging the local guarantees in (25) gives

$$\frac{d}{t}\mathbb{E}_V \|\mathcal{A}_V(q - x_V)\|_2^2 = d\mathbb{E}_{V,I} \|A_I(q - x_V)\|_2^2 \leq \alpha^2 r^2.$$

The equality uses the uniform choice of a block occurrence, and the inequality follows by comparing with $x^\star$ and using the exact norm identity. For each i, put

$$T_i = \|A_i(q - x^\star - a_i)\|_2^2 .$$

Conditioning on $I = i$ decomposes the projected error into T_i and a conditional variance. As in Lemma 5.2, the latter is at least $(1-\tau)/d$ times the corresponding variance in the original space. Thus

$$\sum_i T_i + (1-\tau)\mathcal{V} \leq \alpha^2 r^2. \tag{26}$$

In particular, $\mathcal{V} \leq \alpha^2 r^2/(1-\tau)$.

We next replace the conditional expected locations $x^\star + a_i$ by the unconditional expected location $x^\star + a$. The comparison in Lemma 5.3 gives

$$\sum_i \|A_i(a_i - a)\|_2^2 \leq \mathbb{E}_I \|a_I - a\|_2^2 + 12\tau \mathbb{E}_V \|x_V - x^\star\|_2^2 .$$

The triangle inequality in each tuple and the local approximation guarantee also give, as in Lemma 5.4,

$$\mathbb{E}_V \|x_V - x^\star\|_2^2 \leq \frac{(1+\alpha)^2}{1-\tau} r^2.$$

Both arguments use only the uniform block frequencies and balance of differences between database points, so they are unchanged for this family.

Finally, decompose the squared distance in the original space:

$$\mathbb{E}_V \|q - x_V\|_2^2 = \sum_i \|A_i(q - x^\star - a)\|_2^2 + \mathcal{V} + \mathbb{E}_I \|a_I - a\|_2^2 .$$

Replacing a by a_i inside the block norms and applying the triangle inequality, exactly as in the exhaustive proof, yields

$$\sqrt{\mathbb{E}_V \|q - x_V\|_2^2} \leq \sqrt{\sum_i T_i + \mathcal{V}} + \sqrt{\sum_i \|A_i(a_i - a)\|_2^2 + \mathbb{E}_I \|a_I - a\|_2^2}.$$

The first term is at most $\alpha r/\sqrt{1-\tau}$ by (26). For the second term, use Lemma 7.2 and the preceding bounds. We obtain

$$\min_v \|q - x_v\|_2 \leq \frac{\alpha + \sqrt{2\alpha^2\rho^2/(1-\rho^2) + 12\tau(1+\alpha)^2}}{\sqrt{1-\tau}}\, r. \tag{27}$$

Here the minimum is at most the root mean squared distance. The bound is the same as for exhaustive tuples, with $1/(s-1)$ replaced by $\rho^2/(1-\rho^2)$. In particular, its approximation factor is $\alpha + O(\rho + \sqrt{\tau})$ when α is bounded and $\rho, \tau \leq 1/2$.

7.4 Proof of the main theorem

Proof of Theorem 1.1. The case $n = 1$ is immediate. For $n \geq 2$, choose $\delta = \varepsilon/100$ and $\tau = \varepsilon^2/10^6$ as in (12), and set

$$\ell = \left\lceil \log_{1/\theta_0}(200/\varepsilon) \right\rceil .$$

Then $\rho \leq \varepsilon/200$ and $\rho^2/(1-\rho^2) \leq \varepsilon^2/10^4$. Substituting these bounds and $\alpha = 1+\delta$ into (27) gives an approximation factor at most $1+\varepsilon$.

It remains to bound the resources. Each tuple has length $t = \Delta^\ell = (1/\varepsilon)^{O(1)}$. Since $n \geq d$, its projected dimension is

$$m = tb = (1/\varepsilon)^{O(1)} \log n.$$

There are d local structures. The deterministic construction of the blocks and Lemma 3.3 therefore give total preprocessing time and space $n^{\mathrm{poly}(1/\varepsilon)}$, including construction of the graph, the tuples, and the projected databases.

On a query, computing the block vectors costs $O(bd \log n)$ time. Assembling and querying each local structure takes time polynomial in m and $\log(n/\delta)$. Using $d \leq n$, the total cost of these steps is $d\,\mathrm{poly}(\varepsilon^{-1}, \log n)$. Finally, comparing the d reported candidates in the original space takes $O(d^2)$ time. Hence the deterministic query time is

$$O\big(d^2 + d\,\mathrm{poly}(\varepsilon^{-1}, \log n)\big) ,$$

as claimed. □

A A deterministic low-dimensional ANN primitive

For completeness, we give the implementation details needed to justify the deterministic bounds in Lemma 3.3.

Proof of Lemma 3.3. We use the reduction from nearest neighbor to near neighbor in Section 2 of [HPIM12], together with the grid construction in Section 3.3. Set both the reduction accuracy and the grid accuracy to $\delta/4$; the resulting approximation factor is at most $(1+\delta/4)^3 \leq 1+\delta$. For Euclidean distance, the grid-volume bound gives $(C/\delta)^{O(m)}$ cells per point, uniformly in m. The resulting space bound is as stated.

We describe how to make the preprocessing deterministic without needing its near-linear expected-time bound. The reduction uses a median connectivity radius: the first radius at which the union of equal-radius balls centered at the points has a connected component containing more than half of them. Compute all pairwise distances and process the edges in nondecreasing order of length. The radius is half the length of the first edge that creates such a component. Thus this radius, and the other connectivity partitions used in preprocessing, can be computed exactly in polynomial time. This replaces the randomized radius-selection routine without changing the recursive decomposition or its correctness.

Finally, store grid-cell keys in deterministic balanced search trees. A comparison between two keys takes $O(m)$ time. If S is the number of keys in a local dictionary, then $\log S = O(m \log(n/\delta))$, so a lookup takes $O(m^2 \log(n/\delta))$ time. The reduction makes $O(\log(n/\delta))$ such lookups and distance computations per query. Keeping $\log(n/\delta)$ in these estimates also covers $\delta < 1/n$. These modifications give the claimed deterministic bounds. □

B Expanders on a power of two vertices

We justify Lemma 7.1 without changing the power-of-two dimension assumed throughout the paper. The Margulis–Gabber–Galil graph on $\mathbb{Z}_m^2$ has the following eight neighbor occurrences at (x, y), with arithmetic modulo m:

$$(x \pm 2y, y), \quad (x \pm (2y+1), y), \quad (x, y \pm 2x), \quad (x, y \pm (2x+1)).$$

Each transformation is paired with its inverse, so the graph is undirected and 8-regular, allowing loops and parallel edges. Its normalized adjacency matrix W_m satisfies

$$\left\| W_m \big|_{\mathbf{1}^\perp} \right\|_{2\to 2} \le \frac{5\sqrt{2}}{8} < 1;$$

see [GG81] and the formulation in Section 8 of [HLW06]. Here the norm bounds the absolute values of all nontrivial eigenvalues.

Proof of Lemma 7.1. Take $\Delta = 16$ and $\theta_0 = 5\sqrt{2}/8$. Every power of two d has the form m^2 or $2m^2$, where m is itself a power of two. If $d = m^2$, duplicate every neighbor occurrence of the graph above. The resulting graph has degree 16 and the same normalized adjacency matrix W_m.

If $d = 2m^2$, use the graph on $\mathbb{Z}_m^2 \times \{0, 1\}$ whose neighbors of (v, a) are all pairs (u, b), where u is a neighbor occurrence of v in the original graph and $b \in \{0, 1\}$. Its degree is 16 and its normalized adjacency matrix is

$$W_m \otimes \frac{1}{2} \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}.$$

The two-point averaging matrix has eigenvalues 1 and 0. Hence the nonzero eigenvalues of the product are those of W_m, and its restriction to the mean-zero subspace has norm at most θ_0. For $d = 1$, take the single-vertex graph with 16 loop occurrences; its mean-zero subspace is $\{0\}$.

In each case, the neighbor list of a vertex is computed with a constant number of arithmetic operations. Constructing all lists therefore takes $O(d)$ operations. □

References

[AC09] Nir Ailon and Bernard Chazelle. The fast Johnson–Lindenstrauss transform and approximate nearest neighbors. *SIAM Journal on Computing*, 39(1):302–322, 2009.

[ACW20] Josh Alman, Timothy M. Chan, and Ryan Williams. Faster deterministic and Las Vegas algorithms for offline approximate nearest neighbors in high dimensions. In *Proceedings of the 31st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2020)*, pages 637–649. SIAM, 2020.

[AFWZ95] Noga Alon, Uriel Feige, Avi Wigderson, and David Zuckerman. Derandomized graph products. *Computational Complexity*, 5(1):60–75, 1995.

[Ahl17] Thomas Dybdahl Ahle. Optimal Las Vegas locality sensitive data structures. In *Proceedings of the 58th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 938–949, 2017.

[AI06] Alexandr Andoni and Piotr Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 459–468, 2006.

[AIR18] Alexandr Andoni, Piotr Indyk, and Ilya Razenshteyn. Approximate nearest neighbor search in high dimensions. In *Proceedings of the International Congress of Mathematicians (ICM 2018)*, volume 4, pages 3287–3318, 2018.

[ALRW17] Alexandr Andoni, Thijs Laarhoven, Ilya Razenshteyn, and Erik Waingarten. Optimal hashing-based time-space trade-offs for approximate near neighbors. In *Proceedings of the 28th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 47–66, 2017.

[AR15] Alexandr Andoni and Ilya Razenshteyn. Optimal data-dependent hashing for approximate near neighbors. In *Proceedings of the 47th Annual ACM Symposium on Theory of Computing (STOC)*, pages 793–801, 2015.

[Cla88] Kenneth L. Clarkson. A randomized algorithm for closest-point queries. *SIAM Journal on Computing*, 17(4):830–847, 1988.

[GG81] Ofer Gabber and Zvi Galil. Explicit constructions of linear-sized superconcentrators. *Journal of Computer and System Sciences*, 22(3):407–420, 1981.

[HLW06] Shlomo Hoory, Nathan Linial, and Avi Wigderson. Expander graphs and their applications. *Bulletin of the American Mathematical Society*, 43(4):439–561, 2006.

[HP01] Sariel Har-Peled. A replacement for Voronoi diagrams of near linear size. In *FOCS*, pages 94–103, 2001.

[HPIM12] Sariel Har-Peled, Piotr Indyk, and Rajeev Motwani. Approximate nearest neighbor: Towards removing the curse of dimensionality. *Theory of Computing*, 8:321–350, 2012.

[IM98] Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *STOC*, pages 604–613, 1998.

[Ind00] Piotr Indyk. Dimensionality reduction techniques for proximity problems. In *SODA*, pages 371–378, 2000.

[KKKÓC20] Matti Karppa, Petteri Kaski, Jukka Kohonen, and Padraig Ó Catháin. Explicit correlation amplifiers for finding outlier correlations in deterministic subquadratic time. *Algorithmica*, 82(11):3306–3337, 2020.

[KOR00] Eyal Kushilevitz, Rafail Ostrovsky, and Yuval Rabani. Efficient search for approximate nearest neighbor in high dimensional spaces. *SIAM Journal on Computing*, 30(2):457–474, 2000.

[Mei93] Stefan Meiser. Point location in arrangements of hyperplanes. *Information and Computation*, 106(2):286–303, 1993.

[Pag16] Rasmus Pagh. Locality-sensitive hashing without false negatives. In *Proceedings of the 27th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1–9, 2016.

[Wei22] Alexander Wei. Optimal Las Vegas approximate near neighbors in ℓ_p. *ACM Transactions on Algorithms*, 18(1), 2022.