

Optimal (Euclidean) Metric Compression*

Piotr Indyk[†]
MIT

Tal Wagner[‡]
Microsoft Research

October 8, 2021

Abstract

We study the problem of representing all distances between n points in $\mathbb{R}^d$, with arbitrarily small distortion, using as few bits as possible. We give asymptotically tight bounds for this problem, for Euclidean metrics, for ℓ_1 (a.k.a. Manhattan) metrics, and for general metrics.

Our bounds for Euclidean metrics mark the first improvement over compression schemes based on discretizing the classical dimensionality reduction theorem of Johnson and Lindenstrauss (Contemp. Math. 1984). Since it is known that no better dimension reduction is possible, our results establish that Euclidean metric compression is possible beyond dimension reduction.

1 Introduction

Contemporary datasets are most often represented as points in a high-dimensional space. Many algorithms are based on the distances induced between those points. Thus, distance computation has emerged as a fundamental scalability bottleneck in many large-scale applications, and spurred a large body of research on efficient approximate algorithms. In particular, a typical goal is to design efficient data structures that, after preprocessing a given set of points, can report approximate distances between those points.

An important complexity measure of these data structures is the space they occupy. Small space usage enables storing more points in the main memory for faster access [JDS11], exploiting fast memory-limited devices like GPUs [JDJ17], and facilitating distributed architectures where communication is limited [CBS20], among other benefits. Indeed, a long line of applied research (e.g., [SH09, WTF09, JDS11, JDJ17, SDSJ19], see also Section 1.3) has been able to perform tasks like image classification in unprecedented scales, by designing distance-preserving space-efficient bit encodings of high-dimensional points.

These methods, while empirically successful, are heuristic in nature and do not possess worst-case guarantees on their accuracy. From a theoretical point of view, the problem can be formalized as follows: *What is the minimal amount of space required to represent all distances between the given data points, up to a given relative error?* In the notable case of Euclidean distances, a fundamental compression result is the dimension reduction theorem of Johnson and Lindenstrauss [JL84], which has been refined to a space-efficient bit encoding (often called a *sketch*) in a sequence of well-known

*A preliminary version [IW17], titled “Near-Optimal (Euclidean) Metric Compression”, appeared in Proc. 28th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2017). The present version improves the main result to a tight bound.

[†]indyk@mit.edu

[‡]tal.wagner@gmail.com. Work done while at MIT.

follow-up works [KOR00, Ach03, AMS99, CCFC02]. However, despite these prominent results, it was not known whether these bounds are tight for compression of Euclidean metrics. In this work, we close this gap and obtain improved and tight sketching bounds for Euclidean metrics, as well as for ℓ_1 metrics and general metric spaces.

1.1 Problem definition

The metric sketching problem is defined as follows:

Definition 1.1 (metric sketching). *Let $1 \leq p \leq \infty$ and $0 < \epsilon < 1$. In the ℓ_p -metric sketching problem, we are given a set of n points $x_1, \dots, x_n \in \mathbb{R}^d$ with ℓ_p -distances in the range $[1, \Phi]$. We need to design a pair of algorithms:*

- *Sketching algorithm: given $x_1, \dots, x_n$, it computes a bitstring called a **sketch**.*
- *Estimation algorithm: given the sketch, it can report for every $i, j \in [n]$ a distance estimate $\tilde{E}_{ij}$ such that*

$$(1 - \epsilon)\|x_i - x_j\|_p \leq \tilde{E}_{ij} \leq (1 + \epsilon)\|x_i - x_j\|_p.$$

The goal is to minimize the bit length of the sketch.

Put simply, the goal is to represent all distances between $x_1, \dots, x_n$, up to distortion $1 \pm \epsilon$, using as few bits as possible. The sketching algorithm can be randomized. In that case, we require that with probability $1 - 1/\text{poly}(n)$ it returns a sketch such that the requirement of the estimation algorithm is satisfied for all pairs $i, j \in [n]$ simultaneously. The estimation algorithm is generally deterministic.

We remark that the assumption on the distances being in $[1, \Phi]$ is essentially without loss of generality, by scaling. If $\min_{i \neq j} \|x_i - x_j\| = M$, we can store in the sketch a 2-approximation M' of M and scale all distances down by M' . This increases the total sketch size additively by $O(\log \log M)$ bits. Then, in all bounds below, Φ becomes the *aspect ratio*, which is the ratio of largest to smallest distance in the given point set.

Euclidean metrics. The most notable case is Euclidean metrics, or $p = 2$. For this case, the celebrated Johnson-Lindenstrauss (JL) dimensionality reduction theorem [JL84] enables reducing the dimension of the input point set to $d' = O(\epsilon^{-2} \log n)$. By the recent result of Larsen and Nelson [LN17], this bound is tight. The JL theorem leads to a sketch of size $O(\epsilon^{-2} \log n)$ **machine words** per point. The **bit size** of the sketch generally depends on the numerical range of distances, encompassed by the parameter Φ (a typical setting to consider below is $\Phi = n^{O(1)}$).¹

For example, if the coordinates of the input points are integers in the range $[-\Phi, \Phi]$ (note that in this case the diameter is $O(\sqrt{d}\Phi)$), then the discretized variant of [JL84] due to Achlioptas [Ach03], and related algorithms like AMS sketch [AMS99] and CountSketch [CCFC02, TZ12], yield a sketch size of $O(\epsilon^{-2} \log(n) \log(d\Phi))$ bits per point. More generally, for any point set with diameter Φ (regardless of coordinate representation), the distance sketches of Kushilevitz, Ostrovsky and Rabani [KOR00] yield a sketch size of $O(\epsilon^{-2} \log(n) \log \Phi)$ bits per point. Perhaps surprisingly, prior to our work, it was not known whether this “discretized JL” upper bound is tight for metric sketching, or can be improved much further. We show it is indeed not tight, by proving an improved and

¹We remark that naïvely rounding each coordinate of the dimension-reduced points to its nearest power of $(1 + \epsilon)$ does not yield a valid sketch. For example, consider two coordinates with values t and $t + 1$, where $t = (1 + \epsilon)^i$ for some integer i . The squared difference between them is 1, whereas after rounding it becomes 0, and the distortion is unbounded.

optimal bound of $O(\epsilon^{-2} \log n + \log \log \Phi)$ amortized bits per points. Our result thus establishes that sketching techniques can go beyond dimension reduction in compressing Euclidean metric spaces.

General metrics. The above formulation also captures sketching of *general metric spaces* — that is, the input is any metric space $([n], d)$ with distances between 1 and Φ — since they embed isometrically into ℓ_∞ with dimension $d = n$. Specifically, for every $i \in [n]$, one defines $x_i = (d(i, 1), \dots, d(i, n)) \in \mathbb{R}^n$. It is not hard to see that $d(i, j) = \|x_i - x_j\|_\infty$ for every $i, j \in [n]$. General metric sketching has been studied extensively, under the name *distance oracles* [TZ05], in larger distortion regimes than $1 \pm \epsilon$, see Section 1.3. We provide tight bounds for distortion $1 \pm \epsilon$.

1.2 Our results

We resolve the optimal sketching size with distortion $1 \pm \epsilon$ for several important classes of metrics: Euclidean metrics, ℓ_1 (a.k.a. Manhattan) metrics, and general metrics. We start with our main results for Euclidean metrics.

Theorem 1.2 (Euclidean metric compression). *For ℓ_2 -metric sketching with n points (of arbitrary dimension) and distances in $[1, \Phi]$, $O(\epsilon^{-2} n \log n + n \log \log \Phi)$ bits are sufficient. If the input dimension is $\Omega(\epsilon^{-2} \log n)$, then $\Omega(\epsilon^{-2} n \log n + n \log \log \Phi)$ bits are also necessary.*

The sketching algorithm is randomized and has running time $O(n^{1+\alpha} \log \Phi + nd \log d + \epsilon^{-2} n \cdot \min\{d \log n, \log^3 n\})$, where d is the ambient dimension of the input metric, and $\alpha > 0$ is an arbitrarily small constant.² The estimation algorithm runs in time $O(\epsilon^{-2} \log(n) \log(\epsilon^{-1} \Phi \log n))$.

Theorem 1.2 improves over the best previous bound of $O(\epsilon^{-2} n \log(n) \log \Phi)$, mentioned above. It also strengthens the upper bound of [AK17] for sketching with additive error (see Section 1.3), and resolves an open problem posed by them.

We note that the sketching algorithm in the above theorem is randomized. This means that with probability $1/\text{poly}(n)$, it may output a sketch that distorts the distances by more than a $(1 \pm \epsilon)$ factor. However, this does not affect the sketch size nor the running time.

By known embedding results, both the upper and lower bound in Theorem 1.2 in fact holds for ℓ_p -metrics for every $1 \leq p \leq 2$, including the notable case ℓ_1 . See Section 5.

We proceed to general metric spaces.

Theorem 1.3 (General metric compression). *For general metric sketching with n points and distances in $[1, \Phi]$, $\Theta(n^2 \log(1/\epsilon) + n \log \log \Phi)$ bits are both sufficient and necessary.*

Note that storing all distances exactly in a general metric takes at least $O(n^2 \log \Phi)$ bits. Naïvely, one could round each distance to its nearest power of $(1 + \epsilon)$, which yields a sketch of size $O(n^2 \log(1/\epsilon) + n^2 \log \log \Phi)$ bits. Theorem 1.3 improves the second term to $n \log \log \Phi$. For example, for the goal of reporting a 2-approximation of each distance (i.e., $d(i, j) \leq \tilde{E}_{ij} \leq 2 \cdot d(i, j)$ for all $i, j \in [n]$), where the input distances are polynomially bounded ($\Phi = n^{O(1)}$), we get a tight bound of $\Theta(n^2)$ bits, compared to the naïve bound of $O(n^2 \log \log n)$ bits.

Both of the theorems above are based on a more general upper bound, that holds for all ℓ_p -metrics.

Theorem 1.4 (ℓ_p -metric compression). *Let $1 \leq p \leq \infty$. For ℓ_p -metric sketching with n points in dimension d and distances in $[1, \Phi]$, $O(n(d + \log n) \log(1/\epsilon) + n \log \log \Phi)$ bits are sufficient. The sketching algorithm is deterministic and runs in time $O(n^2 \log \Phi + nd \log(1/\epsilon))$. The estimation algorithm runs in time $O(d \log(d\Phi))$ for $p < \infty$, and $O(d \log \Phi)$ for $p = \infty$.*

²As $\alpha \rightarrow 0$, the sketch size increases as $O(\alpha^{-1} \log(\alpha^{-1}) \cdot \epsilon^{-2} n \log n + n \log \log \Phi)$.

	Reference	Bits per point	No. queries	Query type
Related work	[JL84, Ach03, KOR00, ...]	$O(\log^2 n)$	$q \leq n^{O(1)}$	distances
	[MMMR18, NN19]	$O(\log^2 n)$	any q	distances
	[MWY13]	$\Omega(\log^2 n)$	$q \geq n$	distances
Our approach	Theorem 1.2	$\Theta(\log n)$	—	none
	[IW18]	$\Theta(\log n \cdot \log q)$	$q \leq n$	distances
	[IW18]	$O(\log n + \log q)$	any q	nearest neighbor

Table 1: Bounds on metric compression with n data points and q query points, in a typical regime with relative error $\epsilon = \Omega(1)$, ambient dimension $d = n^{O(1)}$ and diameter $\Phi = n^{O(1)}$.

The upper bound of Theorem 1.3 follows immediately from Theorem 1.4, since as mentioned earlier, general metric spaces with n points embed isometrically into ℓ_∞ with dimension $d = n$. Similarly, for Euclidean metrics, one can apply the Johnson-Lindenstrauss transform as a preprocessing step in order to reduce the dimension of the input points to $O(\epsilon^{-2} \log n)$, and then apply Theorem 1.4. This gives an upper bound looser than that of Theorem 1.2 by $O(\log(1/\epsilon))$. To obtain the tight bound, we will use additional properties special to Euclidean metrics.

1.3 Additional related work

Sketching with additive error. In a work concurrent to our original paper [IW17], Alon and Klartag [AK17] studied a closely related problem of approximating squared Euclidean distances between points of norm at most 1, up to an *additive* error of ϵ (whereas distortion $1 \pm \epsilon$, as in Definition 1.1, is equivalent to *relative* error ϵ). For this problem, they proved a tight sketching bound of $O(\epsilon^{-2} \log n)$ bits per point.³ Further work on additive error refined this result by parameterizing the reduced dimension by complexity measures of the embedded pointsets, designing faster or deterministic algorithms, or introducing sigma-delta-style quantization [DS20, Sto19, HS20, ZS20, PS20].

Sketching with additive error is generally less restrictive than relative error, in the following sense — on one hand, a relative error sketch implies an additive error sketch by setting $\Phi = O(1/\epsilon)$,⁴ and on the other hand, lower bounds for additive error hold for relative error as well. In particular, as we discuss in Section 6, the lower bound from [AK17] (as well as the lower bound in another concurrent paper [LN17]) provides another way to show the lower bound in Theorem 1.2.

New query points and nearest neighbor search. In the model considered in this paper, the query algorithm needs to report distances only between points $x_1, \dots, x_n$ that were fully known to the sketching algorithm (see Definition 1.1). In a closely related but different setting, the query algorithm gets a set of new points $y_1, \dots, y_q \in \mathbb{R}^d$, that were not known to the sketching algorithm,

³Note that in this model, the parameter Φ does not need to enter the sketch size, since the error is allowed to be arbitrarily larger than the minimal distance in the pointset.

⁴To this end, given a pointset $X = \{x_1, \dots, x_n\}$ with norms bounded by 1, let $\mathcal{N}$ be an $\epsilon/2$ -separated ϵ -net of the unit ball (see Section 2 for the definitions). Let $Y = \{y_1, \dots, y_n\}$ be the respective nearest neighbors of X in $\mathcal{N}$. The separation property of $\mathcal{N}$ implies that Y has aspect ratio at most $O(1/\epsilon)$. We may now sketch the distances in Y using a relative error sketch with $\Phi = O(1/\epsilon)$, since given $i, j \in [n]$, reporting the distance between y_i, y_j instead of x_i, x_j increases the additive error by at most $O(\epsilon)$ by the triangle inequality.

and needs to estimate distances between each y_j and each x_i . A notable example of this setting is the nearest neighbor search problem.

The classical dimension reduction approach, which yields a dimension bound of $O(\epsilon^{-2} \log n)$ and a sketching bound of $O(\epsilon^{-2} n \log(n) \log \Phi)$ bits per point, can handle as many as $q = n^{O(1)}$ query points. Very recently, a new line of work known as *terminal dimension reduction* [MMMR18, NN19] was able to obtain the same bounds for an *unbounded* number of query points q . On the other hand, the papers [JW13, MWY13] proved a matching lower bound of $\Omega(\epsilon^{-2} n \log(n) \log \Phi)$ bits for sketching ℓ_1 or ℓ_2 distances, if $q \geq n$, settling the optimal sketch size in this regime.

In a companion work [IW18], we develop the techniques of the current paper and prove nearly tight sketching bounds in the complement regime $q \leq n$, interpolating between Theorem 1.2 and the above tight bounds for $q \geq n$. Furthermore, we show that for the easier task of reporting an approximate nearest neighbor in the dataset for each query point (rather than estimating all distances between dataset points and query points), a better sketching upper bound is possible. The picture is summarized in Table 1.

Applied literature. A prominent line of applied research (including [SH09, WTF09, JDS11, GLGP12, NF13, GHKS13, KA14, JDJ17, SDSJ19]; see also the surveys [WLKC16, WZS⁺18]) has been dedicated to designing empirical solutions to the problem in Definition 1.1, under the label *learning to hash*. This nomenclature reflects the fact that in the preprocessing stage, these methods employ machine learning techniques to adapt the sketches to the given dataset, in order to optimize performance. While empirically successful, these methods are fundamentally heuristic and do not pose formal solutions to Definition 1.1. In a companion work [IRW17], we design a sketch that on one hand has close to optimal worst-case guarantees (in particular, its size lossier than Theorem 1.2 by $O(\log \log n + \log(1/\epsilon))$), while on the other hand it empirically matches or improves the performance of state-of-the-art heuristic methods.

Distance oracles. The distance oracle problem [TZ05] is equivalent to sketching of general metrics, and has been studied in a different distortion regime. A long line of work (including [PS89, ADD⁺93, Mat96, TZ05, WN12, Che15] and more) has shown that for every integer $k \geq 1$, it is possible to compute a sketch of size $\tilde{O}(n^{1+1/k})$ with distortion $2k - 1$, which is tight up to logarithmic factors under the Erdős Girth Conjecture. Notably, for distortion 3 and above, the sketch size is $o(n^2)$. (However, note that in order to achieve a near-linear sketch size, the distortion must be almost logarithmic.) On the other hand, for any distortion less than 3, it is not hard to show (by considering all shortest-path metrics induced by bipartite simple graphs) that a sketch size of $\Omega(n^2)$ is necessary. For distortion $1 \pm \epsilon$, to our knowledge, the best upper bound prior to our work had been $O(n^2(\log \log \Phi + \log(1/\epsilon)))$ bits, which follows from naïve rounding as mentioned earlier.

1.4 Technical overview

The basic strategy in the sketch is to store each point by its relative location to a nearby point which had already been (approximately) stored. Note that this is different than dimension reduction and its discretizations, which approximately store the location of each point in the space in an absolute sense.

More precisely, let $X = \{x_1, \dots, x_n\}$ be the point set we wish to sketch. For every point $x \in X$, we aim to define a *surrogate* $s^*(x) \in \mathbb{R}^d$, which is an approximation of x that can be efficiently stored in the sketch. To this end, we choose an *ingress* point $in(x) \in X$ near x , and define $s^*(x)$ inductively by its location relative to $s^*(in(x))$, namely $s^*(x) = s^*(in(x)) + [x - s^*(in(x))]_\gamma$, where

$[y]_\gamma$ denotes rounding y to a γ -net, with an appropriate precision γ . We then hope to use the distance between the surrogates, $\|s^*(x_i) - s^*(x_j)\|$, as an estimate for the distance $\|x_i - x_j\|$ for all pairs $i, j \in [n]$. The challenge is to choose the ingresses and the precisions in a way that on one hand ensures a small relative error estimate for each pair, while on the other hand does not occupy too many storage bits.

In order to ensure a relative error approximation of every distance, we need to consider all possible distance scales. To this end we construct a hierarchical clustering tree of the metric space, and define the ingresses and surrogates for clusters (or tree nodes) instead of individual points. Here, it may seem natural to use separating decomposition trees such as [Bar96, CCG⁺98, FRT04], which provide both a separating property (far points are in different clusters) and a packing property (close points are often in the same cluster). However, such trees are bound to incur a super-constant gap between the two properties [Bar96, Nao17], which would lead to a suboptimal sketch size. Instead, our tree transitively merges any two clusters within a sufficiently small distance. This yields a perfect separation property, but no packing property — the diameter of each cluster may be unbounded. We replace it by a global bound on all cluster diameters in the tree (Lemma 3.2).

The tree size is first reduced to linear by compressing long non-branching paths. From a distance estimation point of view, this means that if a cluster is very well separated from the rest of the metric, then we can replace it entirely with one representative point (called *center*) for the purpose of estimating the distances between internal and external points. Then, the crucial step is a careful choice of the ingresses, that ensures that if we set the precisions so as to get correct estimates between all pairs, the total sketch occupies sufficiently few bits. This completes the description of the data structure, which we call *relative location tree*.

In order to estimate the distance $\|x_i - x_j\|$ for a given pair $i, j \in [n]$, we can identify in the tree two nodes v_i, v_j , such that (i) the center of v_i is a sufficiently good proxy for x_i from the point of view of x_j , and vice-versa, (ii) the error between the center of v_i and its surrogate is proportional to $\epsilon \cdot \|x_i - x_j\|$, and the same holds for v_j , and (iii) the surrogates of v_i and v_j can be recovered from the sketch (by following ingresses along the tree) up to a shift, which while unknown, is the same for both. Then we may return the distance between the shifted surrogates as the output distance estimate.

Euclidean metrics. The above outline describes our upper bound for sketching ℓ_p -metrics. However, for Euclidean metrics, the resulting sketch size is suboptimal in the dependence on ϵ . To achieve the optimal bound we further develop the sketch.

To this end, we incorporate randomness into the sketching algorithm. To see why this might help, view a surrogate $s^*(x)$ as an estimator for the point x it represents. In the deterministic sketch described above, $s^*(x)$ is necessarily a *biased* estimator, since it is a fixed point close to but different than x . This bias bears on the sketch size: if, say, the surrogate is chosen by deterministically rounding x to its nearest neighbor in a fixed net, then in order to get a desired level of accuracy $\|x - s^*(x)\| \leq \epsilon$, the net must have size $\Theta(1/\epsilon)^d$, and hence the surrogate requires $\Omega(d \log(1/\epsilon))$ bits to store in the sketch. To improve this, we could hope to use an *unbiased* estimator for x by designing a distribution over surrogates, which we call *probabilistic surrogates*.

Alon and Klartag [AK17] took this approach to achieve the optimal sketch size for *additive* error ϵ . By using *randomized* rounding on the net, they showed its size can be reduced to $O(1)^d$, while $\|x - s^*(x)\| \leq \epsilon$ still holds by probabilistic concentration if the dimension is large enough ($d \gtrsim \epsilon^{-2} \log n$). To achieve *relative* error ϵ , we incorporate this into our techniques described above. We build a relative location tree with $\epsilon = \Omega(1)$; this does not exceed the optimal sketch size for Euclidean metrics, but does not provide the desired approximation of distances. We then

augment it with randomized roundings of displacement vectors between nodes to their surrogates, and between centers to non-centers in well-separated clusters. To estimate the distance $\|x_i - x_j\|$ of a given pair $i, j \in [n]$, we sum an appropriate subset of those randomly rounded displacements along the tree, obtaining probabilistic surrogates X_i, X_j . These are random variables with expected values x_i, x_j up to an unknown but equal shift, and with variance appropriately related to $\|x_i - x_j\|$. For technical reasons related to probabilistic independence, we return a proxy of the distance $\|X_i - X_j\|$ rather than the distance itself, and the result is tightly concentrated at the correct value $\|x_i - x_j\|$.

1.5 Paper organization

Section 2 sets up preliminaries and notation. Section 3 contains the description of the main sketch, and proves the upper bound for ℓ_p metrics in Theorem 1.4. (The upper bound for general metrics in Theorem 1.3 follows as a corollary, as explained above.) Section 4 develops the sketch further and proves the upper bound for Euclidean metrics in Theorem 1.2. Section 5 points out that bounds for Euclidean metrics hold for all ℓ_p metrics with $1 \leq p \leq 2$. Section 6 proves the lower bounds in Theorems 1.2 and 1.3.

2 Preliminaries

We start by stating the classical dimension reduction theorem of Johnson and Lindenstrauss [JL84].

Theorem 2.1 ([JL84]). *Let $x_1, \dots, x_n \in \mathbb{R}^d$, $\epsilon, \delta \in (0, 1)$, and $d' \geq c\epsilon^{-2} \log(n/\delta)$ for a sufficiently large constant $c > 0$. There is a distribution over matrices $M \in \mathbb{R}^{d' \times d}$ (for example, i.i.d. entries from $\frac{1}{\sqrt{d'}} \cdot N(0, 1)$) such that with probability $1 - \delta$, for all $i, j \in [n]$,*

$$(1 - \epsilon)\|x_i - x_j\|_2 \leq \|Mx_i - Mx_j\|_2 \leq (1 + \epsilon)\|x_i - x_j\|_2.$$

2.1 Grid nets

Let $1 \leq p \leq \infty$. Let $\mathcal{B}_p^d = \{x \in \mathbb{R}^d : \|x\|_p \leq 1\}$ denote the d -dimensional ℓ_p -unit ball. Let $\gamma > 0$. A subset $\mathcal{N} \subset \mathbb{R}^d$ is called a γ -net of $\mathcal{B}_p^d$ if for every $x \in \mathcal{B}_p^d$ there is $y \in \mathcal{N}$ such that $\|x - y\|_p \leq \gamma$. Further, $\mathcal{N}$ is γ' -separated if the distance between any pair of distinct points in $\mathcal{N}$ is at least γ' . It is a well-known fact that $\mathcal{B}_p^d$ has a γ -net of size $(c/\gamma)^d$ for a constant $c > 0$ that is $\gamma/2$ -separated, and that the size bound is tight up to the constant c .

We will use a specific net, given by the intersection of the ball with an appropriately scaled grid. For $\rho > 0$, let $\mathcal{G}^d[\rho] \subset \mathbb{R}^d$ denote the uniform d -dimensional grid with cell side length ρ . Namely, $\mathcal{G}^d[\rho]$ is defined as the set of points $\mathbb{R}^d$ such that each of their coordinates is an integer multiple of ρ .

The net we use is $\mathcal{N}_\gamma = 2 \cdot \mathcal{B}_p^d \cap \mathcal{G}^d[\gamma/d^{1/p}]$ (where $2 \cdot \mathcal{B}_p^d$ is the origin-centered ball of radius 2). We drop the dependence on d and p from the notation $\mathcal{N}_\gamma$ for simplicity. Also, in the case $p = \infty$, we use $d^{1/p} = 1$ as a convention. Since each cell of $\mathcal{N}_\gamma$ is a hypercube of diameter γ , or part of one, $\mathcal{N}_\gamma$ is indeed a γ -net of $\mathcal{B}_p^d$ (and is $\gamma/d^{1/p}$ -separated). It is also well-known that $|\mathcal{N}_\gamma| \leq (c'/\gamma)^d$ for a constant $c' > 0$ (see, e.g., [HPIM12] or [AK17]), meaning that $\mathcal{N}_\gamma$ attains the optimal size for γ -nets up to the constant c' . Finally, given $x \in \mathcal{B}_p^d$, we can find $y \in \mathcal{N}_\gamma$ such that $\|x - y\|_p \leq \gamma$ by dividing each coordinate by $\gamma/d^{1/p}$, rounding it to the largest smaller integer, and multiplying it by $\gamma/d^{1/p}$. We call this operation *rounding x to $\mathcal{N}_\gamma$* . In summary,

Lemma 2.2. *For every $x \in \mathcal{B}_p^d$, we can round it to $\mathcal{N}_\gamma$ in time $O(d)$, and store the resulting point of $\mathcal{N}_\gamma$ with $O(d \log(1/\gamma))$ bits.*

We also record another variant of the above lemma.

Claim 2.3. *Let $x \in \mathbb{R}^d$ and $\gamma > 0$. The number of points in $\mathcal{G}^d[\gamma/d^{1/p}]$ which are at distance at most 2γ from x (in the ℓ_p -norm distance) is $O(1)^d$.*

3 The relative location tree

In this section we prove Theorem 1.4, which implies the upper bound in Theorem 1.3, and will also serve as a stepping stone toward Theorem 1.2. The sketching scheme is based on a new data structure that we call *relative location tree*.

Let $X = \{x_1, \dots, x_n\} \subset \mathbb{R}^d$ be a given point set endowed with the ℓ_p -metric for a fixed $1 \leq p \leq \infty$, with minimal distance 1 and diameter Φ . We assume w.l.o.g. that Φ is an integer. To simplify notation, we drop the subscript p from ℓ_p -norms (that is, we write $\|x - y\|$ for $\|x - y\|_p$).

3.1 Hierarchical tree construction

We start by building a hierarchical clustering tree T^* over the points X , by the following bottom-up process. In the bottom level, numbered 0, every point x_i forms a singleton cluster $\{x_i\}$. Level $\ell > 0$ is generated from level $\ell - 1$ by merging any clusters at distance less than 2^ℓ , until no such pair remains. (The distance between two clusters $C, C' \subset X$ is defined as $\text{dist}(C, C') = \min_{x \in C, x' \in C'} \|x - x'\|$.) By level $\lceil \log \Phi \rceil$, the pointset has been merged into one cluster, which forms the root of the tree.

For every tree node v in T^* , we denote its level by $\ell(v)$, its associated cluster by $C(v) \subset X$, its cluster diameter by $\Delta(v)$, and its degree (number of children) by $\deg(v)$. For every $x_i \in X$, let $\text{leaf}(x_i)$ denote the tree leaf whose associated cluster is $\{x_i\}$.

Note that the nodes at each level of T^* form a partition of X . On one hand, we have the following separation property.

Claim 3.1. *If x_i, x_j are at different clusters of the partition induced by level ℓ , then $\|x_i - x_j\| \geq 2^\ell$.*

On the other hand, we have the following global bound on the cluster diameters.

Lemma 3.2. $\sum_{v \in T^*} 2^{-\ell(v)} \Delta(v) \leq 4n$.

Proof. We write uv to denote an edge from a parent u to a child v . We call it a *1-edge* if $\deg(u) = 1$, and a *non-1-edge* otherwise. Note that since T^* has n leaves, it has at most $2n$ non-1-edges. We define edge weights and node weights in T^* as follows. The weight of an edge uv is $\text{wt}(uv) = 0$ if uv is a 1-edge, and $\text{wt}(uv) = 2^{\ell(u)}$ otherwise. The weight of a node v , denoted $\text{wt}(v)$, is the sum of all edge weights in the tree under v (that is, $\text{wt}(v) = \sum_{uu'} \text{wt}(uu')$ where the sum is over all edges uu' such that u is reachable from v by a downward path in T^*).

We argue that $\Delta(v) \leq \text{wt}(v)$ for every node v . This is seen by bottom-up induction on T^* . In the base case v is a leaf, and then $\Delta(v) = \text{wt}(v) = 0$. For the induction step, fix a node u and consider two cases. In the first case, u has degree 1 and a single outgoing 1-edge uv . Then $C(u) = C(v)$ by the tree construction, and $\text{wt}(u) = \text{wt}(v)$ since $\text{wt}(uv) = 0$, thus the claim follows by induction. In the second case u has multiple outgoing edges $\{uv_i : i = 1, \dots, k\}$. Since $\{C(v_i) : i = 1, \dots, k\}$ is a partition of $C(u)$, the diameter $\Delta(u)$ is upper-bounded by $\sum_{i=1}^k (\Delta(v_i) + \text{dist}(C(v_i), C(u) \setminus C(v_i)))$. By induction, $\Delta(v_i) \leq \text{wt}(v_i)$ for every i . By the tree construction, $\text{dist}(C(v_i), C(u) \setminus C(v_i)) \leq 2^{\ell(u)} = \text{wt}(uv_i)$. Together, $\Delta(u) \leq \sum_{i=1}^k (\text{wt}(v_i) + \text{wt}(uv_i)) = \text{wt}(u)$, as needed.

Consequently, it now suffices to prove the bound $\sum_{v \in T^*} 2^{-\ell(v)} \text{wt}(v) \leq 4n$. To this end we count the contribution of each edge to the sum. A 1-edge has no contribution since its weight is 0.

For a non-1-edge uv , let $u_0 = u$, and let u_i be the parent of u_{i-1} for all $i > 0$ until the root is reached. Then uv contributes its weight $2^{\ell(u)}$ to $\text{wt}(u_i)$ for every $i \geq 0$, and its total contribution is $\sum_{i \geq 0} 2^{-\ell(u_i)} \cdot 2^{\ell(u)}$. Since $\ell(u_i) = \ell(u) + i$, the latter sum equals $\sum_{i \geq 0} 2^{-i} < 2$. Since T^* has at most $2n$ non-1-edges, the desired bound follows. $\square$

3.1.1 Path compression

Next, we compress long non-branching paths in T^* . A *1-path* in T^* is a downward path $v_0, v_1, \dots, v_k$ such that $v_1, \dots, v_{k-1}$ are degree-1 nodes. It is called *maximal* if v_0 and v_k are not degree-1 nodes (v_k may be a leaf). For every such path in T^* , if

$$k > \log(2^{-\ell(v_k)} \Delta(v_k)/\epsilon), \quad (1)$$

we replace the path from v_1 to v_k with a *long edge* directly connecting v_1 to v_k . We mark it as long and annotate it with the original path length, k . The rest of the edges are called *short edges*. Note that the right-hand side in Equation (1) depends both on the level of v_k in the tree, $\ell(v_k)$, and on the diameter of the cluster it represents in the metric space, $\Delta(v_k)$.

The tree after path compression will be denoted by T . We note that $\ell(v)$ will continue to denote the original level of v in T^* (or equivalently, the level in T if the long edges are counted according to their lengths).

Lemma 3.3. $\sum_{v \in T} \log(2^{-\ell(v)} \Delta(v)) \leq 4n$.

Proof. Follows from Lemma 3.2 since every node v in T is also present in T^* (with the same level $\ell(v)$ and associated cluster diameter $\Delta(v)$), and since $\log(z) < z$ for all $z \in \mathbb{R}$. $\square$

Lemma 3.4. T has at most $2n(2 + \log(1/\epsilon))$ nodes.

Proof. We charge the degree-1 nodes on every maximal 1-path in T to the bottom node of the path. The total number of nodes in T can then be written as $\sum_{v: \deg(v) \neq 1} k(v)$, where $k(v)$ is the length of the maximal 1-path whose bottom node is v . Due to path compression, we have $k(v) \leq \log(2^{-\ell(v)} \Delta(v)) + \log(1/\epsilon)$. Since T has n leaves, it has at most $2n$ nodes whose degree is not 1, so the total contribution of the second term is at most $2n \log(1/\epsilon)$. For the total contribution of the first term, we need to show $\sum_{v: \deg(v) \neq 1} \log(2^{-\ell(v)} \Delta(v)) \leq 4n$. This is given by Lemma 3.3. $\square$

3.1.2 Subtrees

We partition T into *subtrees* by removing the long edges. Let $\mathcal{F}(T)$ denote the set of resulting subtrees. Furthermore let $\mathcal{L}(T)$ denote the set of nodes of T which are leaves of subtrees in $\mathcal{F}(T)$. Note that a node in $\mathcal{L}(T)$ is either a leaf in T or the top node of a long edge in T . These nodes are special in that they represent clusters whose diameter can be bounded individually.

Lemma 3.5. For every $u \in \mathcal{L}(T)$, $\Delta(u) \leq 2^{\ell(u)} \epsilon$.

Proof. If u is a leaf in T then $C(u)$ contains a single point, thus $\Delta(u) = 0$, and the lemma holds. Otherwise, u is the top node of a long edge in T . Let v be the bottom node of that edge. By path compression, the long edge represents a 1-path of length at least $\log(2^{-\ell(v)} \Delta(v)/\epsilon)$ (see Equation (1)), hence $\ell(u) \geq \ell(v) + \log(2^{-\ell(v)} \Delta(v)/\epsilon)$, and hence $2^{\ell(u)} \geq 2^{\ell(v) + \log(2^{-\ell(v)} \Delta(v)/\epsilon)} = \epsilon^{-1} \Delta(v)$. Since no clusters are merged along a 1-path, we have $C(u) = C(v)$, hence $\Delta(u) = \Delta(v)$, and the lemma follows. $\square$

3.2 Tree annotations: Centers, ingresses, and surrogates

We now augment T with the following annotations, which would efficiently encode information on the location of its clusters. Each cluster in the tree is represented by one of its points, chosen largely arbitrarily, called its *center*. The center location is stored using the approximate displacement from a nearby cluster center (already stored by induction), called its *ingress*. The approximate location of the center is called its *surrogate*.

3.2.1 Centers

For every node v in T we choose a *center* from the points in its cluster $C(v)$, in a bottom-up manner, as follows. For a leaf $v = \text{leaf}(x_i)$, let $c(v) = i$. For a non-leaf v with children $u_1, \dots, u_k$, let $c(v) = \min\{c(u_i) : i \in [k]\}$. The point $x_{c(v)}$ is the center of v .

3.2.2 Ingresses

Next, for every node u in T we assign an *ingress* node, denoted $\text{in}(v)$. Intuitively, the ingress is a node in T such that $x_{c(\text{in}(v))}$ is close to $x_{c(v)}$, and our eventual purpose is to store the latter by its location relative to the former.

Before turning to the formal definition of ingresses, let us give an intuitive overview. The distance between $x_{c(v)}$ and $x_{c(\text{in}(v))}$ can generally be as large as $\Delta(v) + \Delta(\text{in}(v))$, since the centers are positioned arbitrarily inside their clusters. Since we plan to store the approximate displacement of $x_{c(v)}$ from $x_{c(\text{in}(v))}$, we would pay the log of that distance in the sketch size. Since we plan to invoke Lemma 3.3 to bound the total size, we can afford to pay the log-diameter of each node only once. This could create a difficulty, since we may wish to use the same node as the ingress for multiple nodes. Our choice of ingresses is meant to avoid this difficulty, by ensuring that $\|x_{c(v)} - x_{c(\text{in}(v))}\|$ depends only on $\Delta(v)$ and not on $\Delta(\text{in}(v))$ (Lemma 3.6 below). To this end, once we have identified a cluster $C(v')$ nearby $C(v)$ at the same tree level, we intuitively want to choose the ingress of v to be not the center of v' , but rather the nearest point to v in $C(v')$. Call that point $x \in C(v')$, and note that $\|x_{c(v)} - x_{c(v')}\|$ could be larger than $\|x_{c(v)} - x\|$ by $\Delta(v')$, which is the term we are trying to avoid. Since ingresses are nodes rather than points, we want $\text{in}(v)$ to be a node whose center is x , ideally $\text{leaf}(x)$, whose diameter is zero. This raises two technical points: One, due to the preceding path compression step, the node $\text{leaf}(x)$ might not be reachable from v' anymore (by short edges), so we instead use the lowest ancestor of $\text{leaf}(x)$ reachable from v' . Two, in order to use $x_{c(\text{in}(v))}$ to approximately store the location of $c(v)$, we need to have already approximately stored $x_{c(\text{in}(v))}$, which means we need an ordering of the nodes such that each node appears after its ingress. We will argue that our somewhat involved choice of ingresses admits such an ordering.

We now formally define the ingresses. They are defined in each subtree $T' \in \mathcal{F}(T)$ separately. For the root r of T' , we set $\text{in}(r) = r$ for convenience (as we will not require ingresses for subtree roots). Now we assign ingresses to all children of every node v in T' , and this would take care of the rest of the nodes in T' . Let $u_1, \dots, u_k$ be the children of v , such that w.l.o.g. $c(v) = c(u_1)$. Consider the simple graph H_v whose nodes are $u_1, \dots, u_k$, where u_i, u_j are neighbors iff $\text{dist}(C(u_i), C(u_j)) \leq 2^{\ell(v)}$. The fact that $C(u_1), \dots, C(u_k)$ have been merged into $C(v)$ in the tree construction means that H_v is a connected graph. Fix an arbitrary spanning tree τ_v of H_v and root it at u_1 . For u_1 , the ingress is $\text{in}(u_1) = v$. For u_i with $i > 1$, let u_j be its parent node in τ_v . Let $x \in C(u_j)$ be the closest point to $C(u_i)$ in $C(u_j)$ (i.e., $x = \arg\min_{x' \in C(u_j)} \min_{x'' \in C(u_i)} \|x' - x''\|$). Let $u_x \in \mathcal{L}(T)$ be the leaf of T' whose cluster contains x . The ingress of u_i is $\text{in}(u_i) = u_x$. See Figure 1 for illustration.

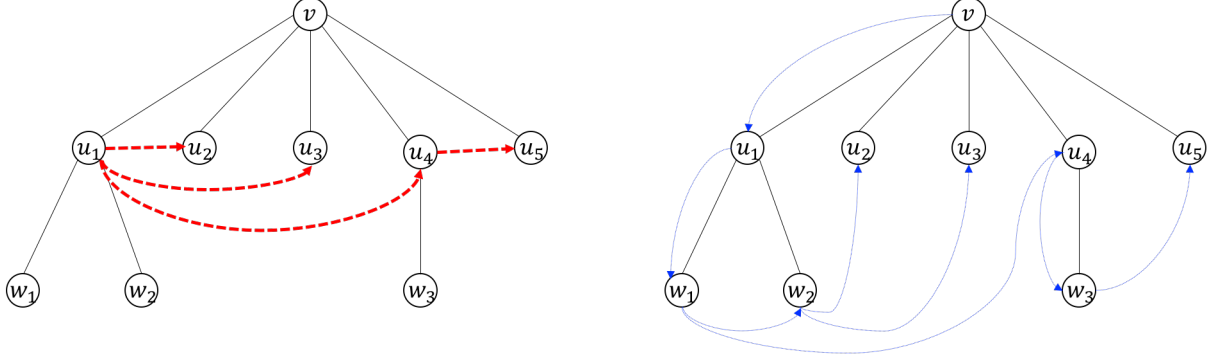


Figure 1: Choice of ingresses. Left: The black tree is a subtree $T' \in \mathcal{F}(T)$, rooted at v , with $c(v) = c(u_1) = c(w_1)$. The dashed red arrows form a spanning tree τ_v on the children of v . Right: The blue dotted arrows denote the ingresses in T' (each arrow source is the ingress of the arrow destination), in the case that $C(w_1)$ contains the point closest to $x_{c(u_4)}$ in $C(u_1)$ (thus w_1 is chosen as the ingress of u_4), and that $C(w_2)$ contains the point closest to $x_{c(u_2)}$ in $C(u_1)$ and the point closest to $x_{c(u_3)}$ in $C(u_1)$ (thus w_2 is chosen as the ingress of u_2 and u_3).

(Note that there is a downward path in T from u_j to $\text{leaf}(x)$, and u_x is the bottom node on that path that belongs to T' . Equivalently, u_x is the bottom node on the path that is reachable from u without traversing a long edge.)

The following lemma bounds the distance from a node center to its ingress center.

Lemma 3.6. *For every node u in T , $\|x_{c(u)} - x_{c(\text{in}(u))}\| \leq 3 \cdot 2^{\ell(u)} + \Delta(u)$.*

Proof. Fix a subtree $T' \in \mathcal{F}(T)$. If u is the root of T' , the claim is obvious since $u = \text{in}(u)$. Next, using the same notation as above, we prove the claim for all children $u_1, \dots, u_k$ of a given node v in T' . For u_1 we have $c(\text{in}(u_1)) = c(v) = c(u_1)$, and the claim holds. For u_i with $i > 1$, recall that u_j denotes its ancestor in τ_v , and that x is a point in $C(u_j)$ that realizes the distance $\text{dist}(C(u_i), C(u_j))$, which is upper-bounded by $2^{\ell(v)}$. Therefore,

$$\|x_{c(u_i)} - x\| \leq \text{dist}(\{x\}, C(u_i)) + \Delta(u_i) \leq 2^{\ell(v)} + \Delta(u_i).$$

Noting that $\ell(v) = \ell(u_i) + 1$, we find

$$\|x_{c(u_i)} - x\| \leq 2 \cdot 2^{\ell(u_i)} + \Delta(u_i). \quad (2)$$

Recall that $\text{in}(u_i)$ was chosen as the leaf in T' whose cluster contains x . In particular, $x_{c(\text{in}(u_i))}$ and x are both contained in $C(\text{in}(u_i))$. By Lemma 3.5, $\|x_{c(\text{in}(u_i))} - x\| \leq 2^{\ell(\text{in}(u_i))}$. Since $\text{in}(u_i)$ is a descendant of a sibling of u_i , we have $\ell(\text{in}(u_i)) \leq \ell(u_i)$, hence $\|x_{c(\text{in}(u_i))} - x\| \leq 2^{\ell(u_i)}$. Combined with Equation (2), this implies the lemma by the triangle inequality. $\square$

We also record the following fact.

Claim 3.7. *For every node u in T , $\ell(\text{in}(u)) \leq \ell(u) + 1$.*

Proof. The ingress is either u itself, the parent of u in T , or a descendant of the parent. $\square$

Ingress ordering. The nodes in every subtree $T' \in \mathcal{F}(T)$ can be ordered such that every node appears after its ingress (except the root, which is its own ingress, and would be first in the ordering). Such ordering is given by a depth-first scan (DFS) on T' , in which additionally, the children of every node v are traversed in a DFS order on τ_v . Since the ingress of every non-root node is either its parent in T' , or a descendant of the sibling in T' which is its predecessor in τ_v , this ordering places every non-root after its ingress as desired. This will be important since the rest of the proof utilizes induction on the ingresses.

3.2.3 Surrogates

Now we can define the surrogates, which are meant to serve as approximate locations for the center of each tree node. We start by defining a *coarse surrogate* $s^*(v)$ for every node v in T . They are defined in every subtree $T' \in \mathcal{F}(T)$ separately, by induction on the ingress order in T' . For the root v of T' , we let $s^*(v) = x_{c(v)}$. For a non-root v in T' , we denote

$$\gamma(v) := \left(5 + \left\lceil \frac{\Delta(v)}{2^{\ell(v)}} \right\rceil\right)^{-1}, \quad (3)$$

and

$$\eta^*(v) = \frac{\gamma(v)}{2^{\ell(v)}} \cdot (x_{c(v)} - s^*(in(v))). \quad (4)$$

Let $\eta(v)$ be the rounding of $\eta^*(v)$ to the grid net $\mathcal{N}_{\gamma(v)}$ (see Section 2). By this we mean that $\eta(v)$ is obtained by rounding each coordinate of $\eta^*(v)$ to the largest smaller integer multiple of $\gamma(v)$. We define $s^*(v)$, by induction on $s^*(in(v))$, as

$$s^*(v) = s^*(in(v)) + \frac{2^{\ell(v)}}{\gamma(v)} \cdot \eta(v). \quad (5)$$

The following lemma bounds the distance between a node center and its surrogate.

Lemma 3.8. *For every v in T , $\|x_{c(v)} - s^*(v)\| \leq 2^{\ell(v)}$.*

Proof. By induction on the ingress ordering in the subtree $T' \in \mathcal{F}(T)$ that contains v . In the base case, v is the root and the claim holds trivially since $s^*(v) = x_{c(v)}$. For a non-root v , we have $\|x_{c(in(v))} - s^*(in(v))\| \leq 2^{\ell(in(v))} \leq 2 \cdot 2^{\ell(v)}$, where the first inequality is by induction on the ingress and the second is by Claim 3.7. By Lemma 3.6 we have $\|x_{c(v)} - x_{c(in(v))}\| \leq 3 \cdot 2^{\ell(v)} + \Delta(v)$, and together, by the triangle inequality, $\|x_{c(v)} - s^*(in(v))\| \leq 5 \cdot 2^{\ell(v)} + \Delta(v)$. By Equations (3) and (4), this implies $\|\eta^*(v)\| \leq 1$. Now, since $\mathcal{N}_{\gamma(v)}$ is a $\gamma(v)$ -net for the unit ball, we have $\|\eta^*(v) - \eta(v)\| \leq \gamma(v)$. Finally,

$$\begin{aligned} \|x_{c(v)} - s^*(v)\| &= \|x_{c(v)} - s^*(in(v)) - \frac{2^{\ell(v)}}{\gamma(v)} \cdot \eta(v)\| && \text{Equation (5)} \\ &= \|x_{c(v)} - s^*(in(v)) - \frac{2^{\ell(v)}}{\gamma(v)} \cdot (\eta^*(v) - \eta^*(v) + \eta(v))\| \\ &= \left\| \frac{2^{\ell(v)}}{\gamma(v)} \cdot (\eta(v) - \eta^*(v)) \right\| && \text{Equation (4)} \\ &\leq 2^{\ell(v)}. \end{aligned}$$

□

3.2.4 Leaf surrogates

For every subtree leaf $v \in \mathcal{L}(T)$ we also use a finer surrogate $s_\epsilon^*(v)$, called *leaf surrogate*. To this end, let $\eta_\epsilon(v)$ be the rounding of $\eta^*(v)$ to the grid net $\mathcal{N}_{\gamma(v) \cdot \epsilon}$, where $\gamma(v)$ and $\eta^*(v)$ are the same as before. The leaf surrogate is defined as

$$s_\epsilon^*(v) = s^*(in(v)) + \frac{2^{\ell(v)}}{\gamma(v)} \cdot \eta_\epsilon(v). \quad (6)$$

Note that $s^*(in(v))$ is the surrogate of $in(v)$ defined earlier (the definition of $s_\epsilon^*(v)$ is not inductive.)

Lemma 3.9. *For every $v \in \mathcal{L}(T)$, $\|x_{c(v)} - s_\epsilon^*(v)\| \leq 2^{\ell(v)} \cdot \epsilon$.*

Proof. The proof of Lemma 3.8 showed that $\|\eta^*(v)\| \leq 1$. Hence, as $\mathcal{N}_{\gamma(v) \cdot \epsilon}$ is a $(\gamma(v) \cdot \epsilon)$ -net for the unit ball, we have $\|\eta^*(v) - \eta_\epsilon(v)\| \leq \gamma(v) \cdot \epsilon$. Thus,

$$\begin{aligned} \|x_{c(v)} - s_\epsilon^*(v)\| &= \|x_{c(v)} - s^*(in(v)) - \frac{2^{\ell(v)}}{\gamma(v)} \cdot \eta_\epsilon(v)\| && \text{Equation (6)} \\ &= \|x_{c(v)} - s^*(in(v)) - \frac{2^{\ell(v)}}{\gamma(v)} \cdot (\eta^*(v) - \eta^*(v) + \eta_\epsilon(v))\| \\ &= \left\| \frac{2^{\ell(v)}}{\gamma(v)} \cdot (\eta_\epsilon(v) - \eta^*(v)) \right\| && \text{Equation (4)} \\ &\leq 2^{\ell(v)} \epsilon. \end{aligned}$$

□

3.3 Sketch size

The sketch stores the tree T , with the following annotations. For each edge we store whether it is long or short, and for the long edges we store their original lengths. For each node v we store the center label $c(v)$, the ingress label $in(v)$, the precision $\gamma(v)$, and the element $\eta(v)$ of the $\gamma(v)$ -net $\mathcal{N}_{\gamma(v)}$. For every node v in $\mathcal{L}(T)$ we also store $\eta_\epsilon(v)$, which is an element of the $(\gamma(v) \cdot \epsilon)$ -net $\mathcal{N}_{\gamma(v) \cdot \epsilon}$. This completes the description of the relative location tree. We now bound the total size of the sketch.

Claim 3.10. (i) T has at most $2n$ long edges. (ii) $|\mathcal{L}(T)| \leq 3n$.

Proof. For part (i), recall that the bottom node of every long edge has degree different than 1. Since T has n leaves, it may have at most $2n$ such nodes. Part (ii) follows from (i) since every leaf of a subtree in $\mathcal{F}(T)$ is either a leaf of T or the top node of a long edge. □

Lemma 3.11. *The total sketch size is $O(n(d + \log n) \log(1/\epsilon) + n \log \log \Phi)$ bits.*

Proof. By Lemma 3.4 we have $|T| = O(n \log(1/\epsilon))$. The tree structure of T can be stored with $O(|T|)$ bits by the Eulerian Tour Technique [TV84]. The length of every long edge is bounded by the number of levels in T , which is $O(\log \Phi)$, and hence by Claim 3.10(i) their total storage cost is $O(n \log \log \Phi)$ bits. The center of each node is an integer in $[n]$, and can be encoded by $\log n$ bits. The ingress of each node is either itself, its parent, or a leaf of a subtree in $\mathcal{F}(T)$, hence by Claim 3.10(ii) it is one of $O(n)$ nodes, and can be encoded by $O(\log n)$ bits. Together, the total storage cost of the centers and ingresses is $O(|T| \log n)$. The total number of bits required to store

the $\gamma(v)$'s is

$$\begin{aligned}
\sum_{v \in T} \log \left(\frac{1}{\gamma(v)} \right) &= \sum_{v \in T} \log \left(5 + \left\lceil \frac{\Delta(v)}{2^{\ell(v)}} \right\rceil \right) \\
&\leq O(|T|) + \sum_{v \in T} \log \left(\frac{\Delta(v)}{2^{\ell(v)}} \right) \\
&\leq O(n \log(1/\epsilon)),
\end{aligned} \tag{7}$$

having used Lemmas 3.3 and 3.4 for the last inequality. Finally, for every node v , $\eta(v)$ is encoded as an element of $\mathcal{N}_{\gamma(v)}$, which by Lemma 2.2 takes $O(d \log(1/\gamma(v)))$ storage bits. Hence by eq. (7), their total storage size is $O(d) \cdot \sum_{v \in T} \log \left(\frac{1}{\gamma(v)} \right) = O(dn \log(1/\epsilon))$ bits. For nodes in $\mathcal{L}(T)$ we also store $\eta_\epsilon(v)$, which is an element in a $(\gamma(v) \cdot \epsilon)$ -net. By Lemma 2.2, this adds $O(d \log(1/\epsilon))$ per node, and by Claim 3.10(ii) there are $O(n)$ such node, hence the total additional cost is $O(nd \log(1/\epsilon))$ bits. Adding up all of the sketch components, the total sketch size is $O(n(d + \log n) \log(1/\epsilon) + n \log \log \Phi)$ bits. $\square$

3.4 Distance estimation

We now show how to use the relative location tree to approximate the distance $\|x_i - x_j\|$ for every pair $i, j \in [n]$. This proves the sketch size bound in Theorem 1.4 (running times are analyzed in the next section). The key point is that within each subtree in $\mathcal{F}(T)$, we can recover the surrogates up to a fixed (unknown) shift from the sketch.

3.4.1 Shifted surrogates

Let $T' \in \mathcal{F}(T)$ be a subtree. For every node v in T' , we define a *shifted surrogate* $s(v) \in \mathbb{R}^d$ by induction on the ingress order in T' , as follows. For the root v of T' , let $s(v) = \mathbf{0}$ (the origin in $\mathbb{R}^d$). For a non-root v in T' , let $s(v) = s(\text{in}(v)) + \frac{2^{\ell(v)}}{\gamma(v)} \cdot \eta(v)$. For $v \in \mathcal{L}(T)$, let the *shifted leaf surrogate* be $s_\epsilon(v) = s(\text{in}(v)) + \frac{2^{\ell(v)}}{\gamma(v)} \cdot \eta_\epsilon(v)$.

Note that we can compute the shifted surrogates from the sketch, since it stores $\text{in}(v)$, $\gamma(v)$ and $\eta(v)$ for every node, and it also stores the lengths of the long edges, which allow us to recover $\ell(v)$. For $v \in \mathcal{L}(T)$ we can compute the shifted leaf surrogate, since the sketch stores $\eta_\epsilon(v)$. Furthermore, observe that the induction step that defines the shifted surrogates is identical to the one defining the surrogates (Equation (5)), and they differ only in the induction base. This implies,

Claim 3.12. *Let v be a node in a subtree $T' \in \mathcal{F}(T)$ whose root is r . Then $s(v) = s^*(v) - x_{c(r)}$. Furthermore, if v is a leaf of T' , then $s_\epsilon(v) = s_\epsilon^*(v) - x_{c(r)}$.*

We remark that $x_{c(r)}$ cannot be recovered for the sketch. Indeed, there can be as many as $\Omega(n)$ subtrees in $\mathcal{F}(T)$, and thus storing all of their root centers could amount to fully (or at least approximately) storing $\Omega(n)$ points — the same problem we are trying to solve.

3.4.2 Estimation algorithm

Given $i, j \in [n]$, we show how to return a $(1 \pm \epsilon)$ -approximation of $\|x_i - x_j\|$. Let u_{ij} be the lowest common ancestor of $\text{leaf}(x_i)$ and $\text{leaf}(x_j)$ in T . Let $T' \in \mathcal{F}(T)$ be the subtree that contains u_{ij} . Let v_i be the leaf of T' whose cluster contains x_i , and similarly define v_j for x_j . See Figure 2 for illustration. The estimate we return is $\|s_\epsilon(v_i) - s_\epsilon(v_j)\|$.

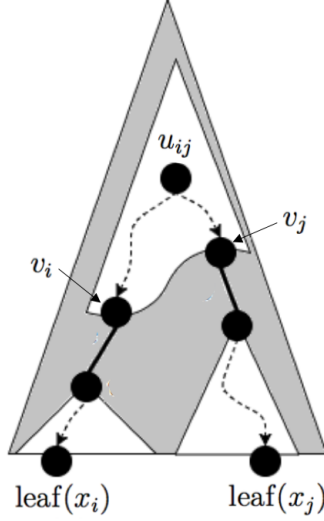


Figure 2: Distance estimation for $\|x_i - x_j\|$. The external shaded triangle is the tree T . The white regions are subtrees. The dashed arrows are downward paths in T . The thick arcs are long edges. The output estimate is $\|s_\epsilon(v_i) - s_\epsilon(v_j)\|$.

Lemma 3.13. $\|s_\epsilon(v_i) - s_\epsilon(v_j)\| = (1 \pm 4\epsilon) \cdot \|x_i - x_j\|$.

Proof. By Claim 3.12, $\|s_\epsilon(v_i) - s_\epsilon(v_j)\| = \|s_\epsilon^*(v_i) - s_\epsilon^*(v_j)\|$. By the triangle inequality,

$$\|s_\epsilon^*(v_i) - s_\epsilon^*(v_j)\| = \|x_i - x_j\| \pm (\|x_i - s_\epsilon^*(v_i)\| + \|x_j - s_\epsilon^*(v_j)\|). \quad (8)$$

Since $v_i \in \mathcal{L}(T)$ and $x_i, x_{c(v_i)} \in C(v_i)$, we have $\|x_i - x_{c(v_i)}\| \leq 2^{\ell(v_i)}\epsilon$ by Lemma 3.5. Combining this with Lemma 3.9 yields $\|x_i - s_\epsilon^*(v_i)\| \leq 2 \cdot 2^{\ell(v_i)}\epsilon$ by the triangle inequality. Since u_{ij} cannot be a leaf in its subtree T' (since then its degree would be either 0 or 1, contradicting its choice as the lowest common ancestor of $\text{leaf}(x_i)$ and $\text{leaf}(x_j)$), we have $\ell(v_i) \leq \ell(u_{ij}) - 1$, and thus $\|x_i - s_\epsilon^*(v_i)\| \leq 2^{\ell(u_{ij})}\epsilon$. The same holds for x_j , and summing these together, $\|x_i - s_\epsilon^*(v_i)\| + \|x_j - s_\epsilon^*(v_j)\| \leq 2 \cdot 2^{\ell(u_{ij})}\epsilon$. By Claim 3.1 $2^{\ell(u_{ij})-1} \leq \|x_i - x_j\|$, and hence $\|x_i - s_\epsilon^*(v_i)\| + \|x_j - s_\epsilon^*(v_j)\| \leq \|x_i - x_j\| \cdot 4\epsilon$. Plugging this into Equation (8) proves the lemma. $\square$

Scaling ϵ by a constant, this concludes the proof of the sketch size bound in Theorem 1.4.

3.5 Running times

Starting with the sketching time, we spend $O(n^2 \log \Phi)$ time constructing T^* . Path compression takes linear time in the size of T^* , which is $O(n \log \Phi)$. To define the ingresses, we need to construct the graph H_v over the children of every node v , and find a spanning tree in it. This takes $O(k_v^2)$ time if v has k_v children. Since in every level ℓ there are up to n nodes, we have $\sum_{v: \ell(v)=\ell} k_v \leq n$, and therefore the total time for level ℓ is $O(\sum_{v: \ell(v)=\ell} k_v^2) \leq O(n^2)$. Over $O(\log \Phi)$ levels in the tree, this too takes $O(n^2 \log \Phi)$ time. Then, for every node v we need to compute $\gamma(v)$, $\eta^*(v)$ and $\eta(v)$ in order to define the surrogates. This involves arithmetic operations on d -dimensional vectors in $O(d)$ time each, as well as rounding $\eta^*(v)$ to a grid net, which by Lemma 2.2 takes $O(d)$ time. Since there are $O(n \log(1/\epsilon))$ nodes (Lemma 3.4), this takes $O(nd \log(1/\epsilon))$ time overall.

We proceed to the estimation time. Since the height of the tree is $O(\log \Phi)$, we spend that much time finding the lowest common ancestor of $\text{leaf}(x_i), \text{leaf}(x_j)$ and finding v_i, v_j . Then we need to compute the shifted leaf surrogates $s_\epsilon(v_i), s_\epsilon(v_j)$. Due to the inductive definition of the surrogates, this might require traversing the ingress ordering on the subtree backward all the way to the root. In the worst case we might traverse all nodes in T , which could take $\Omega(n)$ time.

To avoid this, we can augment the sketch with additional information that improves the query time without asymptotically increasing the sketch size. In particular, we explicitly store the shifted surrogates for some nodes in T , called *landmark nodes*. Let $K = \lceil \log(2\Phi \cdot d^{1/p}) \rceil$. We choose landmark nodes in each subtree $T' \in \mathcal{F}(T)$ separately, as follows: Let T'_{in} be the tree that describes the ingress ordering in T' (this is a tree on the nodes in T' with the same root, where the parent of each node v is $in(v)$). Start with a lowest node $v \in T'_{in}$; climb upward K steps (or less if the root is reached), to a node $\hat{v}$; declare $\hat{v}$ a landmark node, remove it from T'_{in} with all its descendants; iterate. Since every iteration but the last removes at least K nodes from T'_{in} , we finish with at most $O(|T'_{in}|/K)$ landmark nodes. Summing over all subtrees, we have $O(|T|/K)$ landmark nodes in total.

For every landmark node, we explicitly store the shifted surrogate in the sketch. Note that choosing landmark nodes and computing their shifted surrogates require the same time as computing the (non-shifted) surrogates (both involve tracing the ingress ordering in each subtree and processing each node in $O(d)$ time), so they do not asymptotically change the sketching time. Furthermore, computing the shifted surrogate of a given non-landmark node can now be done in $O(dK)$ time. Thus, the total estimation time for $p < \infty$ is $O(d \log(d\Phi))$, and for $p = \infty$ (where $d^{1/p} = 1$) it is $O(d \log \Phi)$.

It remains to see that storing the shifted surrogates for landmark nodes does not asymptotically increase the sketch size. To this end, note that the shifted surrogates are defined recursively, starting at $\mathbf{0}$, and in each step adding a vector of the form $\gamma(v)^{-1} \cdot 2^{\ell(v)} \cdot \eta(v)$. Since $\eta(v)$ is an element in $\mathcal{N}_{\gamma(v)}$ — the grid net with cell side length $\gamma(v)/d^{1/p}$ — every coordinate of a shifted surrogate is an integer multiple of $d^{-1/p}$. On the other hand, we have the following:

Claim 3.14. *For every node v in T , $\|s(v)\| \leq 2\Phi$.*

Proof. Let r be the root of the subtree $T' \in \mathcal{F}(T)$ that contains v . By Claim 3.12, $\|s(v)\| = \|s^*(v) - x_{c(r)}\|$. By the triangle inequality, $\|s^*(v) - x_{c(r)}\| \leq \|s^*(v) - x_{c(v)}\| + \|x_{c(v)} - x_{c(r)}\|$. By Lemma 3.8, $\|s^*(v) - x_{c(v)}\| \leq \Phi$. Since Φ is an upper bound on the diameter of the input point set X , $\|x_{c(v)} - x_{c(r)}\| \leq \Phi$. Together, $\|s(v)\| \leq 2\Phi$. $\square$

The claim implies in particular that each coordinate of a shifted surrogate is at most 2Φ . Being also an integer multiple of $d^{-1/p}$, it can be represented by $\lceil \log(2\Phi \cdot d^{1/p}) \rceil = K$ bits. Thus each shifted surrogate is stored by $O(dK)$ bits, and since we store this for $O(|T|/K)$ landmark nodes, the overall additional cost is $O(d|T|) = O(nd \log(1/\epsilon))$ (Lemma 3.4), which does not asymptotically increase the sketch size.

4 Euclidean metrics

In this section we prove the upper bound in Theorem 1.2. We start with Johnson-Lindenstrauss dimension reduction, Theorem 2.1. By applying the theorem as a preprocessing step before our sketching algorithm, we may henceforth assume that $d = O(\epsilon^{-2} \log n)$. Since we may arbitrarily increase the dimension (by adding zero coordinates), we will also assume w.l.o.g. that $d \geq 3\epsilon^{-2} \log n$.

4.1 Sketch augmentations

In this section we describe the sketch. First, we compute the sketch from Section 3, with, say, $1/2$ instead of ϵ (the choice of constant does not matter). Next, we add some augmentations to the sketch.

Let us give a short overview of them. Our goal is to improve the sketch size from the previous section by a factor of $\log(1/\epsilon)$ for Euclidean metrics. This factor originates in two places where the construction of the relative location tree uses deterministic rounding: (i) in rounding displacements to grid nets to define the surrogates, and (ii) in compressing long 1-paths into long edges (effectively rounding every point in the associated cluster to the cluster center from the point of view of points outside the cluster). The two types of augmentations we now introduce essentially replace these with randomized roundings — *surrogate grid quantization* replaces (i), and *long-edge grid quantization* replaces (ii). Using an inductive argument (Lemma 4.1), we show how to construct from them appropriate *probabilistic surrogates* for every pair of points. In the Euclidean case, they can serve instead of the deterministic surrogates of the previous section, while achieving the desired sketch size.

We now formally define the sketch augmentations. To this end, we choose $2d$ i.i.d. random variables uniformly over $[0, 1]$, and arrange them into two vectors $\sigma', \sigma'' \in \mathbb{R}^d$ that will serve as random shifts. (They will not be stored in the sketch, so there is no concern about the precision of their representation.) We will use uniform grids as defined in Section 2. By the “bottom-left” corner of a grid cell, we mean the point in the cell (considered as a closed set of $\mathbb{R}^d$) in which each coordinate is minimized. That is, the bottom-left corner of the d -dimensional hypercube $[a_1, b_1] \times \dots \times [a_d, b_d]$ is $(a_1, \dots, a_d)$.

Below, let $\sigma \in \{\sigma', \sigma''\}$. Note that $\|\frac{1}{\sqrt{d}}\sigma\| \leq 1$ for all supported σ . For every subtree leaf $v \in \mathcal{L}(T)$, we also store in the sketch the following information.

Augmentation I: Surrogate grid quantization. By Lemma 3.8 we have $\|x_{c(v)} - s^*(v)\| \leq 2^{\ell(v)}$. By the triangle inequality, $\|x_{c(v)} + \frac{1}{\sqrt{d}}2^{\ell(v)}\sigma - s^*(v)\| \leq 2 \cdot 2^{\ell(v)}$. By Claim 2.3, the grid with cell side $\frac{1}{\sqrt{d}}2^{\ell(v)}$ has $\exp(d)$ cells intersecting the origin-centered ball of radius $2 \cdot 2^{\ell(v)}$ (where we use $\exp(d)$ to denote $O(1)^d$). Therefore, with $O(d)$ bits we can store the bottom-left corner of the grid cell containing $x_{c(v)} + \frac{1}{\sqrt{d}}2^{\ell(v)}\sigma - s^*(v)$. Since σ is random, this bottom-left corner is a d -dimensional random variable, which we denote by $A_v = (A_v^1, \dots, A_v^d)$.

Augmentation II: Long-edge grid quantization. If the subtree $T' \in \mathcal{F}(T)$ that contains v also contains the root of T , we do not need to store additional information for v . Otherwise, the root of T' is the bottom node of a long edge. Let u be the top node of that long edge, and note that $u \in \mathcal{L}(T)$. See Figure 3 for illustration.

Since v is a descendant of u in T we have $x_{c(v)} \in C(u)$, and hence by Lemma 3.5, $\|x_{c(v)} - x_{c(u)}\| \leq 2^{\ell(u)}$ (recall we use a relative location tree with $\epsilon = \Omega(1)$). By the triangle inequality, $\|x_{c(v)} + \frac{1}{\sqrt{d}}2^{\ell(u)}\sigma - x_{c(u)}\| \leq 2 \cdot 2^{\ell(u)}$. By Claim 2.3, the grid with cell side $\frac{1}{\sqrt{d}}2^{\ell(u)}$ has $\exp(d)$ cells intersecting the origin-centered ball of radius $2 \cdot 2^{\ell(u)}$. Therefore, with $O(d)$ bits we can store the bottom-left corner of the grid cell containing $x_{c(v)} + \frac{1}{\sqrt{d}}2^{\ell(u)}\sigma - x_{c(u)}$. Since σ is random, this corner is a d -dimensional random variable, which we denote by $B_v = (B_v^1, \dots, B_v^d)$.

Remark. Note that we store each of the above augmentations twice — once with the random shift σ' and once with σ'' . To ease notation, let us not denote them separately, and simply keep in

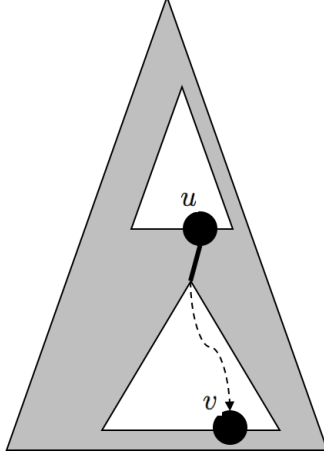


Figure 3: Augmentation to the Euclidean sketch. The external shaded triangle is the tree T . The white regions are subtrees. The thick arc is a long edge. For every subtree leaf v and $\sigma \in \{\sigma' \sigma''\}$, the sketch encodes $x_{c(v)} + \frac{1}{\sqrt{d}} 2^{\ell(v)} \sigma - s^*(v)$ in Augmentation I. If u is defined for v , then the sketch also encodes $x_{c(v)} + \frac{1}{\sqrt{d}} 2^{\ell(u)} \sigma - x_{c(u)}$ in Augmentation II.

mind that we have two independent copies of each A_v and B_v .

Total sketch size. By Lemma 3.11, the relative location tree with $\epsilon = \Omega(1)$ is stored in $O(n(\log n + d + \log \log \Phi))$ bits. The above augmentations store $O(d)$ additional bits per node in $\mathcal{L}(T)$, of which there are $O(n)$ (Claim 3.10), and this does not increase the sketch size asymptotically. Since $d = O(\epsilon^{-2} \log n)$ by the preceding dimension reduction step, the total sketch size is $O(\epsilon^{-2} n \log n + n \log \log \Phi)$ bits.

4.2 Probabilistic surrogates

We now show how to recover, for every point $x \in X$, a random variable that would serve as a probabilistic (shifted) surrogate. For the two next lemmas, fix a subtree $T' \in \mathcal{F}(T)$ with root r . For every $i \in [n]$ such that $x_i \in C(r)$, denote by v_i the leaf of T' whose cluster contains x . That is, v_i is the lowest node on the downward path from r to $\text{leaf}(x_i)$ that does not traverse a long edge.

Lemma 4.1. *Let $i \in [n]$ be such that $x_i \in C(r)$. We can recover from the sketch a d -dimensional random variable $X_i = (X_i^1, \dots, X_i^d) \in \mathbb{R}^d$, such that:*

- *Its coordinates are independent.*
- *Each coordinate is supported on an interval of length at most $\frac{1}{\sqrt{d}} 3 \cdot 2^{\ell(v_i)}$.*
- *$\mathbb{E}[X_i] = x_i - x_{c(r)}$, coordinate-wise.*

We prove this by proving a somewhat more general claim by induction.

Lemma 4.2. *Let $i \in [n]$ be such that $x_i \in C(r)$. For every subtree leaf $v \in \mathcal{L}(T)$ which is a descendant of v_i in T (note that v is not in T' unless $v = v_i$), we can recover from the sketch a d -dimensional random variable $Y_v = (Y_v^1, \dots, Y_v^d) \in \mathbb{R}^d$, such that:*

- *Its coordinates are independent.*

- Each coordinate is supported on an interval of length at most $\frac{1}{\sqrt{d}}(3 \cdot 2^{\ell(v_i)} - 2 \cdot 2^{\ell(v)})$.
- $\mathbb{E}[Y_v] = x_{c(v)} - x_{c(r)}$, coordinate-wise.

Lemma 4.2 clearly implies Lemma 4.1 in the special case $v = \text{leaf}(x_i)$.

Proof of Lemma 4.2. The proof is by induction on the subtree leaves (nodes in $\mathcal{L}(T)$) that lie on the downward path from v_i to $\text{leaf}(x_i)$.

Induction base. In the base case, $v = v_i$. We take $Y_v = A_v + s(v)$. Note that A_v is stored by Augmentation I, and $s(v)$ is a shifted surrogate, so both can be recovered from the sketch. We show that Y_v satisfies the required properties.

Let us simplify some notation for convenience. Let $L = \frac{1}{\sqrt{d}}2^{\ell(v)}$. Let $\mathcal{G}[L]$ be origin-centered uniform grid with cell side length L . Let $y = x_{c(v)} - s^*(v)$, with coordinates $y = (y_1, \dots, y_d)$. Let $H = [a_1, a_1 + L] \times \dots \times [a_d, a_d + L] \subset \mathbb{R}^d$ be the hypercube cell of $\mathcal{G}[L]$ that contains y . Fix $\sigma \in \{\sigma', \sigma''\}$, and let $(\sigma_1, \dots, \sigma_d)$ denote its coordinates.

In Augmentation I, A_v is the bottom-left corner of the cell of $\mathcal{G}[L]$ that contains $y + L\sigma$, where each coordinate of σ is an i.i.d. uniformly random shift in $[0, 1]$. This means that each coordinate $j \in [d]$ of A_v is set to $A_v^j = a_j$ if $y_j + L\sigma_j < a_j + L$, and to $A_v^j = a_j + L$ otherwise. The latter condition rearranges to $\sigma_j < 1 - \frac{1}{L}(y_j - a_j)$ (note that this value is in $[0, 1]$ since a_j is defined such that $a_j \leq y_j < a_j + L$), which occurs with probability $1 - \frac{1}{L}(y_j - a_j)$. Therefore,

$$\mathbb{E}_{\sigma_j}[A_v^j] = a_j \cdot (1 - \frac{1}{L}(y_j - a_j)) + (a_j + L) \cdot \frac{1}{L}(y_j - a_j) = y_j.$$

Furthermore, A_v is supported on the corners of the grid cell H , and hence each coordinate is supported on an interval of length $L = \frac{1}{\sqrt{d}}2^{\ell(v)}$. Finally, since the coordinates of σ are independent, then so are the coordinates of A_v . (This is the same randomized rounding scheme from [AK17]; see Figure 4 for illustration.) By taking $Y_v = A_v + s(v)$, the support length of each coordinate and the independence between the coordinates are preserved, while the expectation changes to

$$\mathbb{E}_{\sigma}[Y_v] = \mathbb{E}_{\sigma}[A_v] + s(v) = y + s(v) = x_{c(v)} - s^*(v) + s(v) = x_{c(v)} - x_{c(r)},$$

coordinate-wise, where we have used Claim 3.12 for the rightmost equality. This proves the base case.

Induction step. Let v be a descendant of v_i , which is different than v_i . Let u be the next node in $\mathcal{L}(T)$ on the upward path from v to v_i . By induction, the statement of Lemma 4.2 holds for u . Therefore we have a random variable Y_u with independent coordinates, each supported on an interval of length $\frac{1}{\sqrt{d}}(3 \cdot 2^{\ell(v_i)} - 2 \cdot 2^{\ell(u)})$, such that $\mathbb{E}[Y_u] = x_{c(u)} - x_{c(r)}$ coordinate-wise.

Augmentation II stores B_v , defined as the bottom-left corner of the cell of the origin-centered grid $\mathcal{G}[\frac{1}{\sqrt{d}}2^{\ell(u)}]$ that contains $x_{c(v)} + \frac{1}{\sqrt{d}}2^{\ell(u)}\sigma - x_{c(u)}$. Similarly to what was shown in the base step for A_v , this implies that $\mathbb{E}_{\sigma}[B_v] = x_{c(v)} - x_{c(u)}$, that B_v has independent coordinates, and that it is supported on the corners of the grid cell that contains $x_{c(v)} - x_{c(u)}$, which means that each coordinate is supported on an interval of length $\frac{1}{\sqrt{d}}2^{\ell(u)}$.

We let $Y_v = Y_u + B_v$. It is easily seen that Y_v has the correct expectation ($\mathbb{E}[Y_v] = x_{c(v)} - x_{c(r)}$ coordinate-wise), that its coordinates are independent, and that each is supported on an interval of length $\frac{1}{\sqrt{d}}(3 \cdot 2^{\ell(v_i)} - 2^{\ell(u)})$. The proof is complete by noticing that $\ell(v) < \ell(u)$, hence $\ell(v) \leq \ell(u) - 1$, hence the length is at most $\frac{1}{\sqrt{d}}(3 \cdot 2^{\ell(v_i)} - 2 \cdot 2^{\ell(v)})$. $\square$

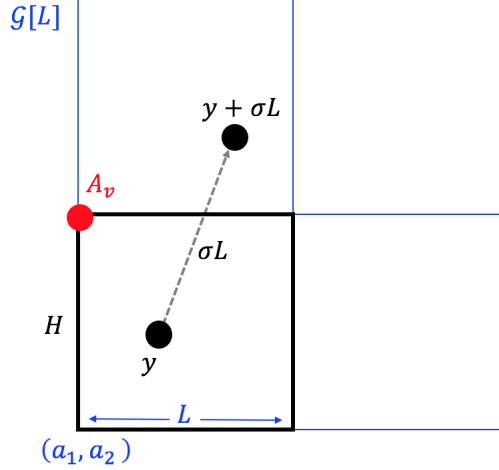


Figure 4: Base case of Lemma 4.2 (in two dimensions). H is the hypercube cell of $\mathcal{G}[L]$ that contains $y = x_{c(v)} - s^*(v)$. Note that H is not random. A_v from Augmentation I is the bottom-left corner of the grid cell that contains $y + \sigma L$, where σ is a shift with uniform i.i.d. coordinates in $[0, 1]$. Thus, A_v is supported on the corners of H , and $\mathbb{E}_\sigma[A_v] = y$.

4.3 Distance estimation

Let $i, j \in [n]$. We show how to estimate $\|x_i - x_j\|$ from the sketch. Let r be the lowest node in T which is the root of a subtree in $T' \in \mathcal{F}(T)$ and such that $x_i, x_j \in C(r)$ (i.e., r is a common ancestor of $\text{leaf}(x_i)$ and $\text{leaf}(x_j)$). Let v_i be the leaf of T' whose cluster contains x_i , and similarly define v_j for x_j . Let $\ell_{ij} := \max\{\ell(v_i), \ell(v_j)\}$. Note that by Claim 3.1 we have $\|x_i - x_j\| \geq 2^{\ell_{ij}}$.

Using Lemma 4.1, we can read off the sketch random variables $X'_i, X'_j, X''_i, X''_j \in \mathbb{R}^d$, such that each has independent coordinates supported on an interval of length $\frac{3}{\sqrt{d}}2^{\ell_{ij}}$, such that $\mathbb{E}[X'_i] = \mathbb{E}[X''_i] = x_i - x_{c(r)}$ and $\mathbb{E}[X'_j] = \mathbb{E}[X''_j] = x_j - x_{c(r)}$ coordinate-wise, and such that (X'_i, X'_j) are independent of (X''_i, X''_j) . The latter property is achieved by using the random shift σ' for (X'_i, X'_j) and the random shift σ'' for (X''_i, X''_j) . The estimate we return is $\sqrt{(X'_i - X'_j)^T (X''_i - X''_j)}$. (Note that if we had $X'_i = X''_i$ and $X'_j = X''_j$ then the estimate would just be $\|X'_i - X'_j\|$; however, we will make use of the independence between (X'_i, X'_j) and (X''_i, X''_j) .) We now show it is a sufficiently accurate estimate.

To this end, let $Z_1 = X'_i - X'_j$ and $Z_2 = X''_i - X''_j$. The returned estimate is $\sqrt{Z_1^T Z_2}$. Note that Z_1, Z_2 are independent, each has independent coordinates supported on an interval of length $\frac{6}{\sqrt{d}}2^{\ell_{ij}}$, and $\mathbb{E}[Z_1] = \mathbb{E}[Z_2] = x - y$ coordinate-wise.

The following lemma is adapted from Alon and Klartag [AK17].

Lemma 4.3. *Suppose $d \geq 3\epsilon^{-2} \log n$. Let $S > 0$. Let $z_1, z_2 \in \mathbb{R}^d$. Let Z_1, Z_2 be d -dimensional independent random variables with independent coordinates, with each coordinate supported on an interval of length $\frac{1}{\sqrt{d}}S$, and such that $\mathbb{E}[Z_1] = z_1$ and $\mathbb{E}[Z_2] = z_2$ coordinate-wise. Then,*

$$\Pr[\|Z_1^T Z_2 - z_1^T z_2\| \leq \epsilon \cdot S(\|z_1\| + \|z_2\| + S)] \geq 1 - \frac{4}{n^3}.$$

Proof. We will denote the coordinates of Z_1 by $(Z_1^1, \dots, Z_1^d)$, and similarly for z_1, Z_2 , and z_2 . By

the triangle inequality,

$$\begin{aligned} |Z_1^T Z_2 - z_1^T z_2| &= |Z_1^T Z_2 - Z_1^T z_2 + Z_1^T z_2 + z_1^T z_2| \\ &\leq |(Z_1 - z_1)^T z_2| + |Z_1^T (Z_2 - z_2)|. \end{aligned}$$

We start with the term $(Z_1 - z_1)^T z_2 = \sum_{i=1}^d z_2^i (Z_1^i - z_1^i)$. By hypothesis, for every coordinate $i \in [d]$ we have $\mathbb{E}[Z_1^i - z_1^i] = 0$ and $|Z_1^i - z_1^i| \leq \frac{1}{\sqrt{d}}S$. Therefore the sum of squares of the summands is upper-bounded by $\frac{1}{d}S^2\|z_2\|^2$. Now by Hoeffding's inequality,

$$\Pr[|(Z_1 - z_1)^T z_2| > \epsilon S \|z_2\|] \leq 2e^{-2\epsilon^2 d} \leq \frac{2}{n^3},$$

where we have used $d \geq 3\epsilon^{-2} \log n$.

We proceed to the term $Z_1^T (Z_2 - z_2) = \sum_{i=1}^d Z_1^i (Z_2^i - z_2^i)$. Again by hypothesis $\mathbb{E}[Z_2^i - z_2^i] = 0$, and since Z_1, Z_2 are independent, $\mathbb{E}[Z_1^i (Z_2^i - z_2^i)] = 0$. The sum of squares is upper-bounded by $\frac{1}{d}S^2\|Z_1\|^2$ as above, and by the triangle inequality, $\|Z_1\| \leq \|z_1\| + \|Z_1 - z_1\| \leq \|z_1\| + S$. Altogether, $\sum_{i=1}^d (Z_1^i (Z_2^i - z_2^i))^2 \leq \frac{1}{d}S^2(\|z_1\| + S)^2$, and by Hoeffding's inequality,

$$\Pr[|Z_1^T (Z_2 - z_2)| > \epsilon S (\|z_1\| + S)] \leq 2e^{-2\epsilon^2 d} \leq \frac{2}{n^3}.$$

The lemma follows by a union bound over the two terms. $\square$

Applying the lemma to Z_1, Z_2 defined above (with $z_1 = z_2 = x_i - x_j$ and $S = 6 \cdot 2^{\ell_{ij}}$),

$$\Pr[|Z_1^T Z_2 - \|x_i - x_j\|^2| \leq \epsilon \cdot 6 \cdot 2^{\ell_{ij}} (2\|x_i - x_j\| + 6 \cdot 2^{\ell_{ij}})] \geq 1 - \frac{4}{n^3}.$$

Since $\|x_i - x_j\| \geq 2^{\ell_{ij}}$, this implies

$$\Pr[|Z_1^T Z_2 - \|x_i - x_j\|^2| \leq \epsilon \cdot 48 \|x_i - x_j\|^2] \geq 1 - \frac{4}{n^3},$$

and thus $Z_1^T Z_2 = (1 \pm O(\epsilon)) \cdot \|x_i - x_j\|^2$, which renders our estimate $\sqrt{Z_1^T Z_2}$ correct (up to scaling ϵ by a constant) with that probability. The total success probability is $1 - O(1/n)$, by a union bound over all pairs $i, j \in [n]$, and over the application of the Johnson-Lindenstrauss theorem that was used as a preprocessing step.

4.4 Running times

The estimation time is as in Theorem 1.4, with dimension $\Theta(\epsilon^{-2} \log n)$. We now focus on the sketching time. The Johnson-Lindenstrauss theorem can be performed either naively in time $O(\epsilon^{-2} nd \log n)$, or in time $O(nd \log d + \epsilon^{-2} n \cdot \min\{d \log n, \log^3 n\})$ by the Fast Johnson-Lindenstrauss Transform of Ailon and Chazelle [AC09]. Note that here, d is the ambient dimension of the input metric (before dimension reduction).

Next we compute the sketch from Section 3. To avoid confusion in notation, let us denote its error and dimension parameters by ϵ' and d' respectively. We construct that sketch with $\epsilon' = \Omega(1)$ and $d' = \Theta(\epsilon^{-2} \log n)$, which as per Section 3.5 takes time $O(n^2 \log \Phi + nd')$. The $n^2 \log \Phi$ term can be reduced to $O(n^{1+\alpha} \log \Phi)$ for any constant $0 < \alpha < 1$, at the cost of increasing the sketch size by an additive factor of $O(nd' \cdot \alpha^{-1} \log(1/\alpha))$. This does not asymptotically increase its size $O(nd' + n \log \log \Phi)$ as long as $\alpha = \Omega(1)$.

To this end, let $c = \alpha^{-1/2}$. In constructing the relative location tree, we use the algorithm of [HPIM12] to compute c -approximate connected components in each level. Their algorithm is based on Locality-Sensitive Hashing (LSH), which in Euclidean spaces can be implemented in time $O(n^{1+1/c^2})$ [AI06]. Using c -approximate connected components means that clusters in level ℓ of the relative location tree may be merged if the distance between them is up to $c \cdot 2^\ell$ (rather than just 2^ℓ), and to account for this constant loss, we need to scale ϵ' down to ϵ'/c . Since the dependence of the sketch size on ϵ' is $O(nd' \log(1/\epsilon'))$ where in our case $d' = \Theta(\epsilon^{-2} \log n)$, it increases by an additive factor of $O(nd' \cdot \alpha^{-1} \log(1/\alpha))$.

Finally, the sketch augmentations in Section 4.1 take time d per node in $\mathcal{L}(T)$ to compute, so in total, $O(nd)$ time. The overall sketching time is as stated in Theorem 1.2.

5 ℓ_p -Metrics with $1 \leq p < 2$

We point out that by known embedding results, both the upper and lower bounds for Euclidean metric compression in Theorem 1.2 apply more generally to ℓ_p -metrics for every $1 \leq p \leq 2$.

Theorem 5.1. *Let $1 \leq p \leq 2$ and $\epsilon > 0$. For ℓ_p -metric sketching with n points and diameter Φ (of arbitrary dimension), $\Theta(\epsilon^{-2} n \log n + n \log \log \Phi)$ bits are both sufficient and necessary.*

The upper bound relies on the well-known fact that every such metric embeds isometrically into a negative-type metric, i.e., into a squared Euclidean metric. We use the following constructive version of this fact, from [LN14, Theorem 116], based on [MN04].

Theorem 5.2 ([LN14]⁵). *Let $1 \leq p < 2$. Let $X \subset \mathbb{R}^d$ be a point set with ℓ_p -aspect ratio Φ . There is a mapping $f : X \rightarrow \mathbb{R}^{d \cdot \text{poly}(\log \Phi, \log d, 1/\epsilon)}$ such that for every $x, y \in X$,*

$$(1 - \epsilon) \|x - y\|_p^p \leq \|f(x) - f(y)\|_2^2 \leq (1 + \epsilon) \|x - y\|_p^p.$$

Proof of Theorem 5.1. Both the upper and lower bound follow from Theorem 1.2. For the upper bound, by Theorem 5.2 we have a map f such that it suffices to report $\|f(x_i) - f(x_j)\|_2^{2/p}$ for every i, j . Then it suffices to sketch the Euclidean metric on $f(x_1), \dots, f(x_n)$. The lower bound follows from the standard fact that Euclidean metrics embed isometrically into ℓ_p -metrics for every $1 \leq p < 2$ (see, e.g., [Mat13]). $\square$

6 Lower bounds

In this section we prove tight compression lower bounds for Euclidean metric spaces and for general metric spaces, matching the upper bounds in Theorems 1.2 and 1.3 respectively. This finishes the proofs of those two theorems.

We start with the lower bound for Euclidean metric sketching. We note that an $\Omega(\epsilon^{-2} n \log n)$ lower bound is also given in [LN17] and [AK17], which appeared concurrently to the original publication of our work [IW17]. The lower bound construction is also similar in all those works. However, since their lower bounds are proven for a less restrictive sketching problem (essentially, an additive approximation of the inner products, rather than a relative approximation as in our case; see Section 1.3), their proofs are considerably more involved than the argument we give below.

⁵The statement in [LN14] is for $\Phi = d^{O(1)}$, but applies to any $\Phi > 0$. The statement given here is by setting $R = d^{-1/q}$ in [LN14, Theorem 116] and scaling the minimal distance in the given metric to 1.

Theorem 6.1 (Euclidean metrics). *The ℓ_2 -metric sketching problem with n points, distances in $[1, \Phi]$ and dimension $d = \Omega(\epsilon^{-2} \log n)$ requires $\Omega(\epsilon^{-2} n \log n + n \log \log \Phi)$ bits.*

Proof. We start by proving the first term of the lower bound, $\Omega(\epsilon^{-2} n \log n)$. Let $0 < \gamma < 0.5$ be a constant and let $\epsilon \geq \Omega(1/n^{0.5-\gamma})$ be smaller than a sufficiently small constant. Let $k = 1/\epsilon^2$, and suppose w.l.o.g. k is an integer by scaling ϵ down by an appropriate constant. Note that $k = O(n^{1-2\gamma}) \ll n$ since $\epsilon \geq \Omega(1/n^{0.5-\gamma})$.

Let B be the set of standard basis vectors in $\mathbb{R}^n$. Let $a_1, \dots, a_n$ be an arbitrary distinct vectors in $\{0, 1\}^n$, each having exactly k coordinates set to 1 (and the rest to 0). Let $A = \{\frac{1}{\sqrt{k}} a_i : i \in [n]\}$. Note that $A \cup B$ is a set of $2n$ points in $\mathbb{R}^n$, each with unit norm.

Suppose we have a sketch for the Euclidean distances in $A \cup B$ up to distortion $1 \pm \frac{1}{8}\epsilon$. This means it can report the squared Euclidean distances up to distortion $1 \pm \frac{1}{2}\epsilon$ (by simply squaring its output). For every $i, j \in [n]$, denote by $a_i(j)$ the j -th coordinate of a_i , or equivalently, $a_i(j) = a_i^T e_j$. Then,

$$\|\frac{1}{\sqrt{k}} a_i - e_j\|_2^2 = \|\frac{1}{\sqrt{k}} a_i\|_2^2 - \frac{2}{\sqrt{k}} a_i^T e_j + \|e_j\|_2^2 = 2 - 2\epsilon a_i(j).$$

Thus, if $a_i(j) = 0$ then $\|\frac{1}{\sqrt{k}} a_i - e_j\|_2^2 = 2$, and the sketch is guaranteed to return at least $2 - \epsilon$. Conversely, if $a_i(j) = 1$ then $\|\frac{1}{\sqrt{k}} a_i - e_j\|_2^2 = 2 - 2\epsilon$, and the sketch is guaranteed to return at most $(1 + \frac{1}{2}\epsilon)(2 - 2\epsilon) = 2 - \epsilon - \epsilon^2$. Consequently, we can recover every $a_i(j)$ from the sketch, and thus recover A . The number of possible choices for A is $\binom{n}{k}$, which by a known estimate $\binom{m}{\ell} \geq (\frac{m}{\ell})^\ell$ for all integers m, ℓ is at least $((\frac{n}{k})^k / n)^n$. Therefore, the resulting bit lower bound on the sketch size is

$$\log \left(\left(\frac{\binom{n}{k}^k}{n} \right)^n \right) = nk \log \left(\frac{n}{k} \right) - n \log n = \frac{n}{\epsilon^2} \cdot \log(n\epsilon^2) \geq \Omega(\gamma \cdot \epsilon^{-2} n \log n),$$

where the final bound is since $\log(n\epsilon^2) \geq \Omega(\log(n^{2\gamma})) = \Omega(\gamma \log n)$, and since we can make ϵ small enough such that $\epsilon^2 < \gamma$. Note that the dimension of the point sets constructed above can be reduced to $O(\epsilon^{-2} \log n)$ by the Johnson-Lindenstrauss theorem [JL84]. This proves the first term of the lower bound in the theorem statement.

Next we prove the second term of the lower bound, $\Omega(n \log \log \Phi)$. Suppose w.l.o.g. that $\log \Phi$ is an integer. Consider the point set $X = \{1, \dots, n\}$. Define a map $g : X \rightarrow \mathbb{R}$ by setting $g(1) = 0$, and for every $x \in X \setminus \{1\}$ setting $g(x) = 2^{\phi(x)}$ with an arbitrary $\phi(x) \in \{1, \dots, \log \Phi\}$. The number of choices for g is $(\log \Phi)^{n-1}$, and every choice of g is a Euclidean embedding of X with one-dimension and aspect ratio at most Φ . We can fully recover g given a Euclidean distance sketch for X with distortion better than 2, since $2^{\phi(x)} = |g(x) - g(1)| = \|g(x) - g(1)\|_2$ for every $x \in X$, and every two possible values of $\phi(x)$ are separated by at least a factor of 2. This yields a sketching bound of $\log((\log \Phi)^{n-1}) = \Omega(n \log \log \Phi)$ bits. $\square$

Next is our lower bound for general metric sketching.

Theorem 6.2 (general metrics). *The general metric sketching problem with n points and distances in $[1, \Phi]$ requires $\Omega(n^2 \log(1/\epsilon) + n \log \log \Phi)$ bits.*

Proof. Let $\epsilon > 0$ be smaller than a sufficiently small constant. We suppose w.l.o.g. that ϵ^{-1} is an integer. We construct a metric space (X, d) with $X = \{1, \dots, n\}$. For every $x, y \in X$ such that $x < y$, set $d(x, y) = 1 + k(x, y) \cdot \epsilon$, with an arbitrary integer $k(x, y) \in \{0, 1, \dots, \epsilon^{-1} - 1\}$. Note that $1 \leq d(x, y) < 2$ for all x, y . This defines a metric space regardless of the choice of the $k(x, y)$'s. Indeed, we only need to verify the triangle inequality, and it holds trivially since

all pairwise distances are lower-bounded by 1 and upper-bounded by 2. Hence we have defined a family of $(1/\epsilon)^{\binom{n}{2}}$ metrics.

Next, observe that a sketch with distortion $(1 \pm \frac{1}{4}\epsilon)$ is sufficient to fully recover a metric from this family. Indeed, for every $x, y \in X$, the sketch is guaranteed to report $d(x, y)$ up to an additive error of $\frac{1}{4}\epsilon \cdot d(x, y)$, which is less than $\frac{1}{2}\epsilon$, while the minimum difference between every pair of possible distances is ϵ by construction. By scaling ϵ by a constant, this proves a lower bound of $\log \left((1/\epsilon)^{\binom{n}{2}} \right) = \Omega(n^2 \log(1/\epsilon))$ on the sketch size in bits. The second lower bound term $\Omega(n \log \log \Phi)$ is by the same proof as Theorem 6.1. $\square$

Acknowledgements. This research was supported in part by NSF awards IIS-144747 and DMS-2022448, MADALGO and Simons Foundation.

References

- [AC09] Nir Ailon and Bernard Chazelle, *The fast johnson–lindenstrauss transform and approximate nearest neighbors*, SIAM J. Comput. **39** (2009), no. 1, 302–322.
- [Ach03] Dimitris Achlioptas, *Database-friendly random projections: Johnson-lindenstrauss with binary coins*, J. Comput. Syst. Sci. **66** (2003), no. 4, 671–687.
- [ADD⁺93] Ingo Althöfer, Gautam Das, David Dobkin, Deborah Joseph, and José Soares, *On sparse spanners of weighted graphs*, Discrete & Computational Geometry **9** (1993), no. 1, 81–100.
- [AI06] Alexandr Andoni and Piotr Indyk, *Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions*, 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2006), 21-24 October 2006, Berkeley, California, USA, Proceedings, 2006, pp. 459–468.
- [AK17] Noga Alon and Bo’az Klartag, *Optimal compression of approximate euclidean distances*, ArXiv preprint arXiv:1610.00239. In IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS) (2017).
- [AMS99] Noga Alon, Yossi Matias, and Mario Szegedy, *The space complexity of approximating the frequency moments*, Journal of Computer and system sciences **58** (1999), no. 1, 137–147.
- [Bar96] Yair Bartal, *Probabilistic approximation of metric spaces and its algorithmic applications*, Foundations of Computer Science, 1996. Proceedings., 37th Annual Symposium on, IEEE, 1996, pp. 184–193.
- [CBS20] Benjamin Coleman, Richard G Baraniuk, and Anshumali Shrivastava, *Sub-linear memory sketches for near neighbor search on streaming data*, ICML, 2020.
- [CCFC02] Moses Charikar, Kevin Chen, and Martin Farach-Colton, *Finding frequent items in data streams*, International Colloquium on Automata, Languages, and Programming, Springer, 2002, pp. 693–703.
- [CCG⁺98] Moses Charikar, Chandra Chekuri, Ashish Goel, Sudipto Guha, and Serge Plotkin, *Approximating a finite metric by a small number of tree metrics*, Proceedings 39th Annual Symposium on Foundations of Computer Science (Cat. No. 98CB36280), IEEE, 1998, pp. 379–388.
- [Che15] Shiri Chechik, *Approximate distance oracles with improved bounds*, Proceedings of the forty-seventh annual ACM symposium on Theory of Computing, 2015, pp. 1–10.
- [DS20] Sjoerd Dirksen and Alexander Stollenwerk, *Binarized johnson-lindenstrauss embeddings*, arXiv preprint arXiv:2009.08320 (2020).
- [FRT04] Jittat Fakcharoenphol, Satish Rao, and Kunal Talwar, *A tight bound on approximating arbitrary metrics by tree metrics*, Journal of Computer and System Sciences **69** (2004), no. 3, 485–497.

- [GHKS13] Tiezheng Ge, Kaiming He, Qifa Ke, and Jian Sun, *Optimized product quantization for approximate nearest neighbor search*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 2946–2953.
- [GLGP12] Yunchao Gong, Svetlana Lazebnik, Albert Gordo, and Florent Perronnin, *Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval*, IEEE transactions on pattern analysis and machine intelligence **35** (2012), no. 12, 2916–2929.
- [HPIM12] Sarel Har-Peled, Piotr Indyk, and Rajeev Motwani, *Approximate nearest neighbor: Towards removing the curse of dimensionality*, Theory of Computing **8** (2012), no. 14, 321–350.
- [HS20] Thang Huynh and Rayan Saab, *Fast binary embeddings and quantized compressed sensing with structured matrices*, Communications on Pure and Applied Mathematics **73** (2020), no. 1, 110–149.
- [IRW17] Piotr Indyk, Ilya Razenshteyn, and Tal Wagner, *Practical data-dependent metric compression with provable guarantees*, Advances in Neural Information Processing Systems, 2017, pp. 2614–2623.
- [IW17] Piotr Indyk and Tal Wagner, *Near-optimal (euclidean) metric compression*, ArXiv preprint arXiv:1609.06295. Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), SIAM, 2017, pp. 710–723.
- [IW18] Piotr Indyk and Tal Wagner, *Approximate nearest neighbors in limited space*, Conference On Learning Theory, 2018, pp. 2012–2036.
- [JDJ17] Jeff Johnson, Matthijs Douze, and Hervé Jégou, *Billion-scale similarity search with gpus*, arXiv preprint arXiv:1702.08734 (2017).
- [JDS11] Herve Jegou, Matthijs Douze, and Cordelia Schmid, *Product quantization for nearest neighbor search*, IEEE transactions on pattern analysis and machine intelligence **33** (2011), no. 1, 117–128.
- [JL84] William B Johnson and Joram Lindenstrauss, *Extensions of lipschitz mappings into a hilbert space*, Contemporary mathematics **26** (1984), no. 189-206, 1–1.
- [JW13] Thathachar S Jayram and David P Woodruff, *Optimal bounds for johnson-lindenstrauss transforms and streaming problems with subconstant error*, ACM Transactions on Algorithms (TALG) **9** (2013), no. 3, 26.
- [KA14] Yannis Kalantidis and Yannis Avrithis, *Locally optimized product quantization for approximate nearest neighbor search*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 2321–2328.
- [KOR00] Eyal Kushilevitz, Rafail Ostrovsky, and Yuval Rabani, *Efficient search for approximate nearest neighbor in high dimensional spaces*, SIAM Journal on Computing **30** (2000), no. 2, 457–474.
- [LN14] Huy Le Nguyen, *Algorithms for high dimensional data*, Ph.D. thesis, Princeton University, 2014.
- [LN17] Kasper Green Larsen and Jelani Nelson, *Optimality of the johnson-lindenstrauss lemma*, ArXiv preprint arXiv:1609.02094. 2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS), IEEE, 2017, pp. 633–638.
- [Mat96] Jiří Matoušek, *On the distortion required for embedding finite metric spaces into normed spaces*, Israel Journal of Mathematics **93** (1996), no. 1, 333–344.
- [Mat13] Jiri Matoušek, *Lecture notes on metric embeddings*, Tech. report, 2013.
- [MMMR18] Sepideh Mahabadi, Konstantin Makarychev, Yury Makarychev, and Ilya Razenshteyn, *Nonlinear dimension reduction via outer bi-lipschitz extensions*, Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, 2018, pp. 1088–1101.
- [MN04] Manor Mendel and Assaf Naor, *Euclidean quotients of finite metric spaces*, Advances in Mathematics **189** (2004), no. 2, 451–494.

- [MWY13] Marco Molinaro, David P Woodruff, and Grigory Yaroslavtsev, *Beating the direct sum theorem in communication complexity with implications for sketching*, Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, Society for Industrial and Applied Mathematics, 2013, pp. 1738–1756.
- [Nao17] Assaf Naor, *Probabilistic clustering of high dimensional norms*, Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, 2017, pp. 690–709.
- [NF13] Mohammad Norouzi and David J Fleet, *Cartesian k-means*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 3017–3024.
- [NN19] Shyam Narayanan and Jelani Nelson, *Optimal terminal dimensionality reduction in euclidean space*, Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, 2019, pp. 1064–1069.
- [PS89] David Peleg and Alejandro A Schäffer, *Graph spanners*, Journal of graph theory **13** (1989), no. 1, 99–116.
- [PS20] Rasmus Pagh and Johan Sivertsen, *The space complexity of inner product filters*, 23rd International Conference on Database Theory (ICDT 2020), Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020.
- [SDSJ19] Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, and Hervé Jégou, *Spreading vectors for similarity search*, 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019, OpenReview.net, 2019.
- [SH09] Ruslan Salakhutdinov and Geoffrey Hinton, *Semantic hashing*, International Journal of Approximate Reasoning **50** (2009), no. 7, 969–978.
- [Sto19] Alexander Stollenwerk, *One-bit compressed sensing and fast binary embeddings.*, Ph.D. thesis, RWTH Aachen University, Germany, 2019.
- [TV84] R.E. Tarjan and U. Vishkin, *Finding biconnected components and computing tree functions in logarithmic parallel time*, 25th Annual Symposium on Foundations of Computer Science, 1984., 1984, pp. 12–20.
- [TZ05] Mikkel Thorup and Uri Zwick, *Approximate distance oracles*, Journal of the ACM (JACM) **52** (2005), no. 1, 1–24.
- [TZ12] Mikkel Thorup and Yin Zhang, *Tabulation-based 5-independent hashing with applications to linear probing and second moment estimation*, SIAM Journal on Computing **41** (2012), no. 2, 293–331.
- [WLKC16] Jun Wang, Wei Liu, Sanjiv Kumar, and Shih-Fu Chang, *Learning to hash for indexing big data: a survey*, Proceedings of the IEEE **104** (2016), no. 1, 34–57.
- [WN12] Christian Wulff-Nilsen, *Approximate distance oracles with improved preprocessing time*, Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms, SIAM, 2012, pp. 202–208.
- [WTF09] Yair Weiss, Antonio Torralba, and Rob Fergus, *Spectral hashing*, Advances in neural information processing systems, 2009, pp. 1753–1760.
- [WZS⁺18] Jingdong Wang, Ting Zhang, Nicu Sebe, Heng Tao Shen, et al., *A survey on learning to hash*, IEEE Transactions on Pattern Analysis and Machine Intelligence **40** (2018), no. 4, 769–790.
- [ZS20] Jinjie Zhang and Rayan Saab, *Faster binary embeddings for preserving euclidean distances*, arXiv preprint arXiv:2010.00712 (2020).