A New Near-linear Time Algorithm For k-Nearest Neighbor Search Using a Compressed Cover Tree

Yury Elkin [1] **Vitaliy Kurlin** [1]

Abstract

Given a reference set R of n points and a query set Q of m points in a metric space, this paper studies an important problem of finding k-nearest neighbors of every point $q \in Q$ in the set R in a near-linear time. In the paper at ICML 2006, Beygelzimer, Kakade, and Langford introduced a cover tree on R and attempted to prove that this tree can be built in $O(n \log n)$ time while the nearest neighbor search can be done in $O(n \log m)$ time with a hidden dimensionality factor. This paper fills a substantial gap in the past proofs of time complexity by defining a simpler compressed cover tree on the reference set R. The first new algorithm constructs a compressed cover tree in $O(n \log n)$ time. The second new algorithm finds all k-nearest neighbors of all points from Q using a compressed cover tree in time $O(m(k + \log n) \log k)$ with a hidden dimensionality factor depending on point distributions of the given sets R, Q but not on their sizes.

1. The Neighbor Search, Overview Of Results

In the modern formulation, the k-nearest neighbor problem is to find all $k \geq 1$ nearest neighbors in a given reference set R for all points from another given query set Q.

Both sets belong to a common ambient space X with a distance metric d satisfying all metric axioms. The simplest example of X is $\mathbb{R}^n$ with the Euclidean metric. A query set Q can be a point or a finite subset of a reference set R.

The *exact* k-nearest neighbor problem asks for all true (exact) k-nearest neighbors in R for every point $q \in Q$.

Another (probabilistic) version of the k-nearest neighbor search Har-Peled & Mendel (2006); Manocha & Girolami (2007) aims to find exact k-nearest neighbors with a given probability. The approximate version Arya & Mount (1993); Krauthgamer & Lee (2004); Andoni et al. (2018); Wang et al. (2021) of the nearest neighbor search looks for an ϵ-approximate neighbor $r \in R$ of every query point $q \in Q$ such that $d(q, r) \leq (1 + \epsilon)d(q, \mathrm{NN}(q))$, where $\epsilon > 0$ is fixed and $\mathrm{NN}(q)$ is the exact first nearest neighbor of q.

Definition 1.1 (diameter and aspect ratio). *For any finite set R with a metric d, the* diameter *is* $\mathrm{diam}(R) = \max_{p \in R} \max_{q \in R} d(p, q)$. *The* aspect ratio *is* $\Delta(R) = \dfrac{\mathrm{diam}(R)}{d_{\min}(R)}$, *where $d_{\min}(R)$ is the shortest distance between points of R.*

Definition 1.2 (k-nearest neighbor set NN_k). *For any point $q \in Q$, let $d_1 \leq \cdots \leq d_{|R|}$ be ordered distances from q to all points of a reference set R whose size (number of points) is denoted by $|R|$. For any $k \geq 1$, the k-nearest neighbor set $\mathrm{NN}_k(q; R)$ consists of all $u \in R$ with $d(q, u) \leq d_k$.*

For $Q = R = \{0, 1, 2, 3\}$, the point $q = 1$ has ordered distances $d_1 = 0 < d_2 = 1 = d_3 < d_4 = 2$. The nearest neighbor sets are $\mathrm{NN}_1(1; R) = \{1\}$, $\mathrm{NN}_2(1; R) = \{0, 1, 2\} = \mathrm{NN}_3(1; R)$, $\mathrm{NN}_4(1; R) = R$. So 0 can be a 2nd neighbor of 1, then 2 becomes a 3rd neighbor of 1, or these neighbors of 0 can be found in a different order.

Problem 1.3 (all k-nearest neighbors search). *Let Q, R be finite subsets of query and reference sets in a metric space (X, d). For any fixed $k \geq 1$, design an algorithm to exactly find k distinct points from $\mathrm{NN}_k(q; R)$ for all $q \in Q$ so that the parametrized worst-case time complexity is near-linear in time $\max\{|Q|, |R|\}$, where hidden constants may depend on structures of Q, R but not on their sizes $|Q|, |R|$.*

In a metric space, let $\bar{B}(p, t)$ be the closed ball with a center p and a radius $t \geq 0$. The notation $|\bar{B}(p, t)|$ denotes the number (if finite) of points in the closed ball. Definition 1.4 recalls the expansion constant c from Beygelzimer et al. (2006a) and introduces the new minimized expansion constant c_m, which is a discrete analog of the doubling dimension Cole & Gottlieb (2006).

Definition 1.4 (expansion constants c and c_m). *A subset R of a metric space (X, d) is called* locally finite *if the set*

[1]Department of Computer Science, University of Liverpool, Liverpool, United Kingdom. Correspondence to: Vitaliy Kurlin <vitaliy.kurlin@gmail.com>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

$\bar{B}(p,t) \cap R$ is finite for all $p \in X$ and $t \in \mathbb{R}_+$. Let R be a locally finite set in a metric space X.

The expansion constant $c(R)$ *is the smallest $c(R) \geq 2$ such that $|\bar{B}(p,2t)| \leq c(R) \cdot |\bar{B}(p,t)|$ for any point $p \in R$ and $t \geq 0$, see Beygelzimer et al. (2006a).*

Introduce the new minimized expansion constant $c_m(R) = \lim\limits_{\xi \to 0^+} \inf\limits_{R \subseteq A \subseteq X} \sup\limits_{p \in A, t > \xi} \dfrac{|\bar{B}(p,2t) \cap A|}{|\bar{B}(p,t) \cap A|}$, *where A is a locally finite set which covers R.*

Lemma 1.5. *For any finite sets $R \subseteq U$ in a metric space, we have that $c_m(R) \leq c_m(U)$ and $c_m(R) \leq c(R)$.*

Note that both $c(R), c_m(R)$ are always defined when R is finite. We show below that a single outlier can make the expansion constant $c(R)$ as large as $O(|R|)$.

In the Euclidean line $\mathbb{R}$, The set $R = \{1, 2, \dots, n, 2n+1\}$ of $|R| = n+1$ points has $c(R) = n+1$ because $\bar{B}(2n+1; n) = \{2n+1\}$ is a single point, while $\bar{B}(2n+1; 2n) = R$ is the full set of $n+1$ points. On the other hand, the same set R can be extended to a larger uniform set $A = \{1, 2, \dots, 2n-1, 2n\}$ whose expansion constant is $c(A) = 2$. So the minimized constant of the original set R is much smaller: $c_m(R) \leq c(A) = 2 < c(R) = n+1$.

The constant c from Beygelzimer et al. (2006a) equals $2^{\dim_{KR}}$ from Krauthgamer & Lee (2004, Section 2.1).

In Krauthgamer & Lee (2004, Section 1.1) the doubling dimension $2^{\dim}$ is defined as a minimum value ρ such that any set X can be covered by 2^{ρ} sets whose diameters are half of the diameter of X. The past work Krauthgamer & Lee (2004) proves that $2^{\dim} \leq 2^n$ for any subset of $\mathbb{R}^n$.

Theorem C.15 in appendix C will prove that $c_m(R) \leq 2^n$ for any a finite subset $R \subset \mathbb{R}^n$, so $c_m(R)$ mimics $2^{\dim}$.

Navigating nets. In 2004, Krauthgamer & Lee (2004, Theorem 2.7) claimed that a navigating net can be constructed in time $O\big(2^{O(\dim_{KR}(R))}|R|(\log|R|)\log(\log|R|)\big)$ and all k-nearest neighbors of a query point q can be found in time $O(2^{O(\dim_{KR}(R \cup \{q\})}(k + \log|R|)$, where $\dim_{KR}(R \cup \{q\})$ is the expansion constant defined above. The paper above sketched a proof of Krauthgamer & Lee (2004, Theorem 2.7) in one sentence and skipped pseudo-codes. Unfortunately, the authors didn't reply to our request for these details.

Modified navigating nets Cole & Gottlieb (2006) were used in 2006 to claim the time $O(\log(n) + (1/\epsilon)^{O(1)})$ to find $(1+\epsilon)$-approximate neighbors. The proof and pseudo-code were skipped for this claim and for the construction of the modified navigating net for the claimed time $O(|R| \cdot \log(|R|))$.

Cover trees. In 2006, Beygelzimer et al. (2006a) introduced a cover tree inspired by the navigating nets Krauthgamer & Lee (2004). This cover tree was designed to prove a worst-case time for the nearest neighbor search in terms of the size $|R|$ of a reference set R and the expansion constant $c(R)$ from Definition 1.4. Assume that a cover tree is already constructed on the set R. Then Beygelzimer et al. (2006a, Theorem 5) claimed that a nearest neighbor of any query point $q \in Q$ could be found in time $O(c(R)^{12} \cdot \log|R|)$.

In 2015, Curtin (2015, Section 5.3) pointed out that the proof of Beygelzimer et al. (2006a, Theorem 5) contains a crucial gap, now also confirmed by a specific example in Elkin & Kurlin (2022a, Counterexample 5.2).

The time complexity result of the cover tree construction algorithm Beygelzimer et al. (2006a, Theorem 6) had a similar issue, the gap of which is exposed rigorously in Elkin & Kurlin (2022a, Counterexample 4.2).

Further studies in cover trees. A noteworthy paper on cover trees Kollar (2006) introduced a new probabilistic algorithm for the nearest neighbor search, as well as corrected the pseudo-code of the cover tree construction algorithm of Beygelzimer et al. (2006a, Algorithm 2). Later in 2015, a new, more efficient implementation of cover tree was introduced in Izbicki & Shelton (2015). However, no new time-complexity results were proven.

Another study Jahanseir & Sheehy (2016) explored connections between modified navigating nets Cole & Gottlieb (2006) and cover trees Beygelzimer et al. (2006a).

Several papers Beygelzimer et al. (2006b); Ram et al. (2009); Curtin et al. (2015) studied the possibility of solving k-nearest neighbor Problem 1.3 by using cover trees on both sets Q, R, see Elkin & Kurlin (2022a, Section 6).

New contributions. This work corrects the past gaps of the single-tree approach Beygelzimer et al. (2006a), which were discovered in Elkin & Kurlin (2022a) by using a new compressed cover tree $\mathcal{T}(R)$ from Definition 2.1, which can be constructed on any finite reference set R with a metric d.

- Definition 2.1 introduces a new compressed cover tree.
- Theorem 3.6 and Corollary 3.10 estimate the time to build a compressed cover tree, which corrects the proof of Beygelzimer et al. (2006a, Theorem 6).
- Theorem 4.9 and Corollary 4.7 estimate the time to find all k-nearest neighbors as in Problem 1.3. These advances correct and generalize Beygelzimer et al. (2006a, Theorem 5).

Table 1. Building data structures with hidden $c_m(R)$ or dimensionality constant 2^{dim} Krauthgamer & Lee (2004, Section 1.1).

Data structure	claimed time complexity	space	proofs
Navigating nets Krauthgamer & Lee (2004)	$O\big(2^{O(\text{dim})} \cdot \lvert R\rvert \cdot \log(\Delta) \cdot \log(\log((\Delta))\big)$	$O(2^{O(\text{dim})}\lvert R\rvert)$	Krauthgamer & Lee (2004, Theorem 2.5)
Compressed cover tree [Definition 2.1]	$O\big(c_m(R)^{O(1)} \cdot \lvert R\rvert \log(\Delta(R))\big)$	$O(\lvert R\rvert)$ Lemma B.1	Theorem 3.6

Table 2. Results for building data structures with the hidden classical expansion constant $c(R)$ of Definition 1.4 or KR-type constant $2^{\text{dim}_{KR}}$ Krauthgamer & Lee (2004, Section 2.1).

Data structure	claimed time complexity	space	proofs
Navigating nets Krauthgamer & Lee (2004)	$O\big(2^{O(\text{dim}_{KR})} \cdot \lvert R\rvert \log(\lvert R\rvert) \log(\log \lvert R\rvert)\big)$, Krauthgamer & Lee (2004, Theorem 2.6)	$O(2^{O(\text{dim})}\lvert R\rvert)$	Not available
Cover tree Beygelzimer et al. (2006a)	$O(c(R)^{O(1)} \cdot \lvert R\rvert \cdot \log \lvert R\rvert)$, Beygelzimer et al. (2006a, Theorem 6)	$O(\lvert R\rvert)$	Elkin & Kurlin (2022a, Counterexample 4.2) shows that the past proof is incorrect
Compressed cover tree [Definition 2.1]	$O\big(c(R)^{O(1)} \cdot \lvert R\rvert \cdot \log \lvert R\rvert\big)$	$O(\lvert R\rvert)$ Lemma B.1	Corollary 3.10

Table 3. Results for exact k-nearest neighbors of one query point $q \in X$ using the hidden classical expansion constant $c(R)$ of Definition 1.4 or KR-type constant $2^{\text{dim}_{KR}}$ Krauthgamer & Lee (2004, Section 2.1) and assuming that all data structures are already built. Note that the dimensionality factor $2^{\text{dim}_{KR}}$ is equivalent to $c(R)^{O(1)}$.

Data structure	claimed time complexity	space	proofs
Navigating nets Krauthgamer & Lee (2004)	$O\big(2^{O(\text{dim}_{KR})}(\log(\lvert R\rvert) + k)\big)$ for $k \geq 1$ Krauthgamer & Lee (2004, Theorem 2.7)	$O(2^{O(\text{dim})}\lvert R\rvert)$	Not available
Cover tree Beygelzimer et al. (2006a)	$O\big(c(R)^{O(1)} \log \lvert R\rvert\big)$ for $k = 1$ Beygelzimer et al. (2006a, Theorem 5)	$O(\lvert R\rvert)$	Elkin & Kurlin (2022a, Counterexample 5.2) shows that the past proof is incorrect
Compressed cover tree, Definition 2.1	$O\big(c(R \cup \{q\})^{O(1)} \cdot \log(k) \cdot (\log(\lvert R\rvert) + k)\big)$	$O(\lvert R\rvert)$, Lemma B.1	Theorem 4.9

Table 4. Results for exact k-nearest neighbors of one point q using hidden $c_m(R)$ or dimensionality constant 2^{dim} Krauthgamer & Lee (2004, Section 1.1) assuming that all structures are built.

Data structure	claimed time complexity	space	proofs
Navigating nets Krauthgamer & Lee (2004)	$O\big(2^{O(\text{dim})} \cdot \log(\Delta) + \lvert\bar{B}(q, O(d(q,R))\rvert\big)$ for $k = 1$	$O(2^{O(\text{dim})}\lvert R\rvert)$	a proof outline in Krauthgamer & Lee (2004, Theorem 2.3)
Compressed cover tree, Definition 2.1	$O\big(\log(k) \cdot (c_m(R)^{O(1)} \log(\lvert\Delta\rvert) + \lvert\bar{B}(q, O(d_k(q,R))\rvert)\big)$	$O(\lvert R\rvert)$, Lemma B.1	Corollary 4.7

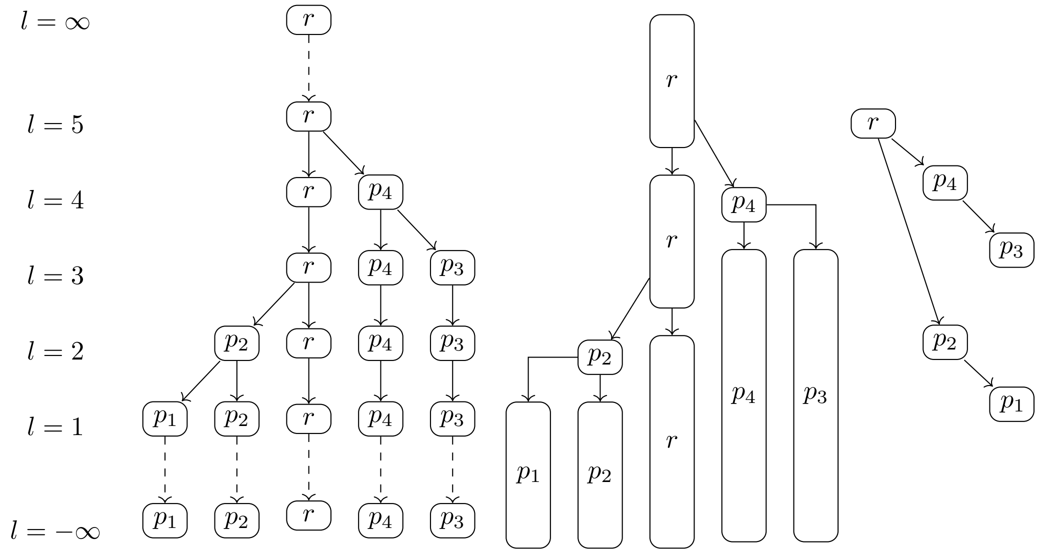

Figure 1. A comparison of past cover trees and a new compressed cover tree in Example B.3. **Left:** an implicit cover tree contains infinite repetitions. **Middle:** an explicit cover tree. **Right:** a compressed cover tree from Definition 2.1 includes each given point exactly once.

2. A New Compressed Cover Tree

This section introduces in Definition 2.1 a new compressed cover tree to solve Problem 1.3. We also prove relevant properties of the expansion constant $c(R)$ and minimized expansion constant $c_m(R)$ of Definition 1.4. All extra details and proofs of this section are in Appendices B,D.

A compressed cover tree in Definition 2.1 will be significantly simpler than an explicit cover tree Elkin & Kurlin (2022a, Definition 2.2), where any given point p can appear in many different nodes simultaneously.

To regain the functionality of the explicit cover tree, we introduce the new concept of a distinctive descendant set $\mathcal{S}_i(p, \mathcal{T}(R))$ in Definition 2.8. See Figure 1 for a comparison between implicit, explicit, and compressed cover trees.

Definition 2.1 (a compressed cover tree $\mathcal{T}(R)$)**.** *Let R be a finite set in a metric space (X, d).* A compressed cover tree *$\mathcal{T}(R)$ has the vertex set R with a root $r \in R$ and a* level *function $l : R \to \mathbb{Z}$ satisfying the conditions below.*

(2.1a) Root condition *: the level of the root node r satisfies $l(r) \geq 1 + \max\limits_{p \in R \setminus \{r\}} l(p)$.*

(2.1b) Cover condition *: for every node $q \in R \setminus \{r\}$, we select a unique* parent *p and a level $l(q)$ such that $d(q, p) \leq 2^{l(q)+1}$ and $l(q) < l(p)$; this parent node p has a single link to its* child *node q.*

(2.1c) Separation condition *: for $i \in \mathbb{Z}$, the* cover set *$C_i = \{p \in R \mid l(p) \geq i\}$ has $d_{\min}(C_i) = \min\limits_{p \in C_i} \min\limits_{q \in C_i \setminus \{p\}} d(p, q) > 2^i$.*

Since there is a 1-1 map between R and all nodes of $\mathcal{T}(R)$, the same notation p can refer to a point in the set R or to a node of the tree $\mathcal{T}(R)$. Set $l_{\max} = 1 + \max\limits_{p \in R \setminus \{r\}} l(p)$ and $l_{\min} = \min\limits_{p \in R} l(p)$. For any node $p \in \mathcal{T}(R)$, $\mathrm{Children}(p)$ denotes the set of all children of p, including p itself. For any node $p \in \mathcal{T}(R)$, define the node-to-root *path as a unique sequence of nodes $w_0, \dots, w_m$ such that $w_0 = p$, w_m is the root and w_{j+1} is the parent of w_j for $j = 0, ..., m-1$.*

A node $q \in \mathcal{T}(R)$ is a descendant *of a node p if p is in the node-to-root path of q. A node p is an* ancestor *of q if q is in the node-to-root path of p. Let $\mathrm{Descendants}(p)$ be the set of all descendants of p, including itself p.*

Lemma 2.2 links the minimized expansion constant with the doubling dimension. This result is used in the proofs of the width bound of a compressed cover tree in Lemma 2.3, also for the time complexity of a compressed cover tree construction in Lemma 3.3, and for the k-nearest neighbor search in Lemma 4.5. All hyperlinks are clickable.

Lemma 2.2 (packing)**.** *Let S be a finite δ-sparse set in a metric space (X, d), so $d(a, b) > \delta$ for all $a, b \in S$. Then, for any point $p \in X$ and any radius $t > \delta$, we have $|\bar{B}(p, t) \cap S| \leq (c_m(S))^{\mu}$, where $\mu = \lceil \log_2(\frac{4t}{\delta} + 1) \rceil$.*

Proof of Lemma 2.2 is in Appendix B.

Lemma 2.3 shows that the number of children of any node of a compressed cover tree on any specific level can be bounded by using minimized expansion constant $c_m(R)$.

Lemma 2.3 (width bound)**.** *Let R be a finite subset of a metric space (X, d). For any compressed cover tree $\mathcal{T}(R)$,*

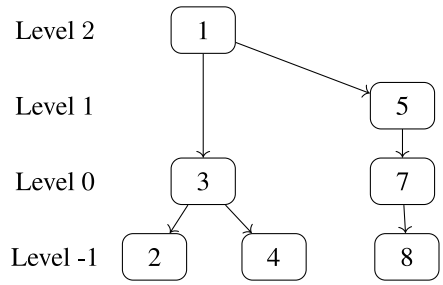

Figure 2. Consider a compressed cover tree $\mathcal{T}(R)$ that was built on set $R = \{1, 2, 3, 4, 5, 7, 8\}$. Let $\mathcal{S}_i(p, \mathcal{T}(R))$ be a distinctive descendant set of Definition 2.8. Then $V_2(1) = \emptyset, V_1(1) = \{5\}$ and $V_0(1) = \{3, 5, 7\}$. And also $\mathcal{S}_2(1, \mathcal{T}(R)) = \{1, 2, 3, 4, 5, 7, 8\}$, $\mathcal{S}_1(1, \mathcal{T}(R)) = \{1, 2, 3, 4\}$ and $\mathcal{S}_0(1, \mathcal{T}(R)) = \{1\}$.

any node p and any level $i \leq l(p)$ we have

$$\{q \in \text{Children}(p) \mid l(q) = i\} \cup \{p\} \leq (c_m(R))^4,$$

where $c_m(R)$ is the minimized expansion constant of R.

Lemma 2.4 is an important property of the expansion constant, which allows us to calculate the low-bound of the number of points in the larger ball $\bar{B}(q, 4r)$ for any node $q \in R$ and radius $r \in \mathbb{R}_+$ using the smaller ball $\bar{B}(q, r)$ and the expansion constant $c(R)$, if there exists a point $p \in R$ which is located in annulus $2r < d(p, q) \leq 3r$.

Lemma 2.4 (growth bound). *Let (A, d) be a finite metric space, let $q \in A$ be an arbitrary point and let $r \in \mathbb{R}$ be a real number. Let $c(A)$ be the expansion constant from Definition 1.4. If there exists a point $p \in A$ such that $2r < d(p, q) \leq 3r$, then $|\bar{B}(q, 4r)| \geq (1 + \frac{1}{c(A)^2}) \cdot |\bar{B}(q, r)|$.*

Lemma 2.5 is a generalization of Lemma 2.4 and will be used to estimate the number of iterations in compressed cover tree construction algorithm, Lemma 3.8 and in the k-nearest neighbors algorithm, Lemma 4.8.

Lemma 2.5 (extended growth bound)**.** *Let (A, d) be a finite metric space, let $q \in A$ be an arbitrary point. Let $p_1, ..., p_n$ be a sequence of distinct points in R, in such a way that for all $i \in \{2, ..., n\}$ we have $4 \cdot d(p_i, q) \leq d(p_{i+1}, q)$. Then*

$$|\bar{B}(q, \frac{4}{3} \cdot d(q, p_n))| \geq (1 + \frac{1}{c(A)^2})^n \cdot |\bar{B}(q, \frac{1}{3} \cdot d(q, p_1))|.$$

Definition 2.6 (the height of a compressed cover tree)**.** *For a compressed cover tree $\mathcal{T}(R)$ on a finite set R, the* height set *is $H(\mathcal{T}(R)) = \{i \mid C_{i-1} \neq C_i\} \cup \{l_{\max}, l_{\min}\}$. The size $|H(\mathcal{T}(R))|$ of this set is called the* height *of $\mathcal{T}(R)$.*

Lemma 2.7. *Any finite set R has the upper bound $|H(\mathcal{T}(R))| \leq 1 + \log_2(\Delta(R))$.*

Intuitively $\mathcal{S}_i(p, \mathcal{T}(R))$ denotes all the descendants of pair (p, i) in the explicit or implicit cover tree.

Definition 2.8 (Distinctive descendant sets)**.** *Let $R \subseteq X$ be a finite reference set with a compressed cover tree $\mathcal{T}(R)$. For any node $p \in \mathcal{T}(R)$ and level $i \leq l(p) - 1$, set $V_i(p) = \{u \in \text{Descendants}(p) \mid i \leq l(u) \leq l(p) - 1\}$. If $i \geq l(p)$, then set $V_i(p) = \emptyset$. For any level $i \leq l(p)$, the* distinctive descendant set *is $\mathcal{S}_i(p, \mathcal{T}(R)) = \text{Descendants}(p) \setminus \bigcup_{u \in V_i(p)} \text{Descendants}(u)$ and has the size $|\mathcal{S}_i(p, \mathcal{T}(R))|$.*

Lemma 2.9 shows that if $q \in \mathcal{S}_i(p, \mathcal{T}(R))$ then there is a node-to-node path $q = a_0, ..., a_m = p$, so that $l(a_{m-1}) \leq i - 1$.

Lemma 2.9. *Let $R \subseteq X$ be a finite reference set with a cover tree $\mathcal{T}(R)$. In the notations of Definition 2.8, let $p \in \mathcal{T}(R)$ be any node. If $w \in \mathcal{S}_i(p, \mathcal{T}(R))$ then either $w = p$ or there exists $a \in \text{Children}(p) \setminus \{p\}$ such that $l(a) < i$ and $w \in \text{Descendants}(a)$.*

Definition 2.10 explains the concrete implementation of a compressed cover tree.

Definition 2.10 ($\text{Children}(p, i)$ and $\text{Next}(p, i, \mathcal{T}(R))$)**.** *In a compressed cover tree $\mathcal{T}(R)$ on a set R, for any level i and a node $p \in R$, set $\text{Children}(p, i) = \{a \in \text{Children}(p) \mid l(a) = i\}$. Let $\text{Next}(p, i, \mathcal{T}(R))$ be the maximal level j satisfying $j < i$ and $\text{Children}(p, i) \neq \emptyset$. If such level does not exist, we set $j = l_{\min}(\mathcal{T}(R)) - 1$. For every node p, we store its set of children in a linked hash map so that*

(a) *any key i gives access to* $\text{Children}(p, i)$,

(b) $\text{Children}(p, i) \to \text{Children}(p, \text{Next}(p, i, \mathcal{T}(R)))$,

(c) *we can directly access* $\max\{j \mid \text{Children}(p, j) \neq \emptyset\}$.

3. Construction Of a Compressed Cover Tree

This section discusses a construction of a compressed cover tree. New Algorithm 3.4 builds a compressed cover tree by using the Insert() method from Beygelzimer et al. (2006a, Algorithm 2), which was specifically adapted for a compressed cover tree, see details in Appendix E.

The proof of Beygelzimer et al. (2006a, Theorem 6), which estimated the time complexity of Beygelzimer et al. (2006a, Algorithm 2), was shown to be incorrect by Elkin & Kurlin (2022a, Counterexample 4.2). The main contribution of this section estimate the time complexity of Algorithm 3.4:

- Theorem 3.6 bounds the time complexity as $O(c_m(R)^{10} \cdot \log_2(\Delta(R)) \cdot |R|)$ via the minimized expansion constant $c_m(R)$ and the aspect ratio $\Delta(R)$.
- Theorem 3.9 bounds the time complexity as $O(c(R)^{12} \cdot \log_2 |R| \cdot |R|)$ via the expansion constant $c(R)$.

Definition 3.1 (construction iteration set $L(\mathcal{T}(W), p)$)**.** *Let W be a finite subset of a metric space (X, d). Let $\mathcal{T}(W)$ be a cover tree of Definition 2.1 built on W and let $p \in X \setminus W$ be an arbitrary point. Let $L(\mathcal{T}(W), p)$ be the set of all levels i during iterations 5-14 of Algorithm 3.5 launched with the inputs $\mathcal{T}(W), p$. We set*

$$\eta(i) = \min\{t \in L(\mathcal{T}(W), p) \mid t > i\}.$$

Theorem 3.2 (correctness of Algorithm 3.4)**.** *Algorithm 3.4 builds a compressed cover tree in Definition 2.1.*

Lemma 3.3 (time complexity of a key step for $\mathcal{T}(R)$)**.** *Arbitrarily order all points of a finite reference set R in a metric space (X, d) starting from the root: $r = p_1$, $p_2, \dots, p_{|R|}$. Set $W_1 = \{r\}$ and $W_{y+1} = W_y \cup \{p_y\}$ for $y = 1, ..., |R| - 1$. Then Algorithm 3.4 builds a compressed cover tree $\mathcal{T}(R)$ in time*

$$O\Big((c_m(R))^8 \cdot \max_{y=1,\dots,|R|-1} L(\mathcal{T}(W_y), p_y) \cdot |R|\Big),$$

where $c_m(R)$ is the minimized expansion constant from Definition 1.4.

Algorithm 3.4 Building a compressed cover tree $\mathcal{T}(R)$ from Definition 2.1.

1: **Input** : a finite subset R of (X, d), root $r \in R$
2: **Output** : a compressed cover tree $\mathcal{T}(R)$.
3: Build the initial compressed cover tree $\mathcal{T} = \mathcal{T}(\{r\})$ consisting of the root node r by setting $l(r) = +\infty$.
4: **for** $p \in R \setminus \{r\}$ **do**
5: $\quad\mathcal{T} \leftarrow$ run AddPoint($\mathcal{T}, p$), Algorithm 3.5.
6: **end for**
7: For the root r of $\mathcal{T}$ set $l(r) = 1 + \max_{p \in R \setminus \{r\}} l(p)$

Theorem 3.6 (time complexity of $\mathcal{T}(R)$ via aspect ratio)**.** *Let R be a finite subset of a metric space (X, d) having the aspect ratio $\Delta(R)$. Algorithm 3.4 builds a compressed cover tree $\mathcal{T}(R)$ in time $O((c_m(R))^8 \cdot \log_2(\Delta(R)) \cdot |R|)$, where $c_m(R)$ is the minimized expansion constant from Definition 1.4.*

Algorithm 3.5 Building $\mathcal{T}(W \cup \{p\})$ in lines 4-6 of Algorithm 3.4.

1: **Function** AddPoint(a compressed cover tree $\mathcal{T}(W)$ with a root r, a point $p \in X$)
2: **Output** : compressed cover tree $\mathcal{T}(W \cup \{p\})$.
3: Set $i \leftarrow l_{\max}(\mathcal{T}(W)) - 1$ and $\eta(l_{\max} - 1) = l_{\max}$ {If the root r has no children then $i \leftarrow -\infty$}
4: Set $R_{l_{\max}} \leftarrow \{r\}$.
5: **while** $i \geq l_{\min}$ **do**
6: $\quad V = \cup_{q \in R_{\eta(i)}} \{a \in \text{Children}(q) \mid l(a) = i\}$.
7: $\quad$Assign $\mathcal{C}_i(R_{\eta(i)}) \leftarrow R_{\eta(i)} \cup V$.
8: $\quad$Set $R_i = \{a \in \mathcal{C}_i(R_{\eta(i)}) \mid d(p, a) \leq 2^{i+1}\}$
9: $\quad$**if** R_i is empty **then**
10: $\quad\quad$Move to **line** 15.
11: $\quad$**end if**
12: $\quad t = \max_{a \in R_i} \text{Next}(a, i, \mathcal{T}(W))$ {If R_i has no children, then we set $t = l_{\min} - 1$}
13: $\quad\eta(i) \leftarrow i$ and $i \leftarrow t$
14: **end while**
15: Pick $v \in R_{\eta(i)}$ minimizing $d(p, v)$. Set $l(p) = \lfloor \log_2(d(p, v) \rfloor - 1$ and define v to be the parent of p and **exit**.

Proof. In Lemma 3.3, use the bounds from Lemma 2.7:

$$\max_{y=2,\dots,|R|} |L(\mathcal{T}(W_{y-1}), p_y)| \leq H(\mathcal{T}(R)) \leq 1 + \log_2 \Delta(R).$$

□

Lemma 3.7. *Let (X, d) be a metric space and let $W \subseteq X$ be its finite subset. Let $q \in X \setminus W$ be an arbitrary point. Let $i \in L(\mathcal{T}(W), q)$ be arbitrarily iteration of Definition 3.1. Assume that $t = \eta(\eta(i+1))$ is defined. Then there exists $p \in W$ satisfying $2^{i+1} < d(p, q) \leq 2^{t+1}$.*

Lemma 3.8 (Construction iteration bound)**.** *Let A, W be finite subsets of a metric space X satisfying $W \subseteq A \subseteq X$. Take a point $q \in A \setminus W$. Given a compressed cover tree $\mathcal{T}(W)$ on W, Algorithm 3.5 runs lines 5-14 this number of times: $|L(\mathcal{T}(W), q)| = O\big(c(A)^2 \cdot \log_2(|A|)\big)$.*

Outline Proof. Assume that Algorithm 3.5 was launched with parameters $(q, \mathcal{T}(W))$ Lemma 3.7 showed that for any iterations $i \in L(\mathcal{T}(W), q)$, if $t = \eta(\eta(i+1))$ exists, then there exists $p \in W$ which belongs to annulus $\bar{B}(q, 2^{t+1}) \setminus \bar{B}(q, 2^{i+1})$. We can select a subsequence S of iterations $L(\mathcal{T}(W), q)$, in such a way that for every $i \in S$ there exists point $p_i \in \bar{B}(q, 2^{t+1}) \setminus \bar{B}(q, 2^{i+1})$. It can be shown that the size of S selected this way is $12 \cdot |S| \geq |L(\mathcal{T}(W), q)|$

Denote by $P = (p_1, ..., p_n)$ the sequence of points p_i obtained from S. Using Lemma 2.5 we obtain

$$|\bar{B}(q, \frac{4}{3} d(q, p_n))| \geq (1 + \frac{1}{c(R)^2})^n \cdot |\bar{B}(q, \frac{1}{3} d(q, p_1))|$$

which can be written as

$$|A| \geq \frac{|\bar{B}(q, \frac{4}{3} \cdot d(q, p_n))|}{|\bar{B}(q, \frac{1}{3} \cdot d(q, p_1))|} \geq (1 + \frac{1}{c(A)^2})^{|S|}$$

Lemma B.7 gives $c(A)^2 \log(A) \geq |S|$. Combining this with the fact that $12 \cdot |S| \geq |L(\mathcal{T}(W), q)|$ we finally conclude that $|L(\mathcal{T}(W), q)| \leq 12 \cdot c(A)^2 \cdot \log_2(|A|)$. $\square$

Theorem 3.9 (time for $\mathcal{T}(R)$ via expansion constants)**.** *Let R be a finite subset of a metric space (X, d). Let A be a finite subset of X satisfying $R \subseteq A \subseteq X$. Then Algorithm 3.4 builds a compressed cover tree $\mathcal{T}(R)$ in time $O((c_m(R))^8 \cdot c(A)^2 \cdot \log_2(|A|) \cdot |R|)$, see the expansion constants $c(A), c_m(R)$ in Definition 1.4.*

Proof. It follows from Lemmas 3.8 and 3.3. $\square$

Corollary 3.10. *Let R be a finite subset of a metric space (X, d). Then Algorithm 3.4 builds a compressed cover tree $\mathcal{T}(R)$ in time $O((c_m(R))^8 \cdot c(R)^2 \cdot \log_2(|R|)) \cdot |R|)$, where the constants $c(R), c_m(R)$ appeared in Definition 1.4.*

Proof. In Theorem 3.9 set $A = R$. $\square$

4. New k-nearest Neighbor Search Algorithm

This section is motivated by Elkin & Kurlin (2022a, Counterexample 5.2), which showed that the proof of past time complexity claim in Beygelzimer et al. (2006a, Theorem 5) for the nearest neighbor search algorithm contained gaps. For extra details and all proofs, see Appendix F.

The gaps are filled by new Algorithm 4.3 for all k-nearest neighbors, which generalizes and improves the original method in Beygelzimer et al. (2006a, Algorithm 1).

The first improvement is the λ-point in line 7, which helps find all k-nearest neighbors of a given query point for any $k \geq 1$. The second improvement is a new break condition for the loop in line 9. This condition is used in the proof of Lemma 4.8 to conclude that the total number of performed iterations is bounded by $O(c(R)^2 \log(|R|))$ during the whole run-time of the algorithm.

The latter improvement corrects the past gap in proof of Beygelzimer et al. (2006a, Theorem 5) by bounding the number of iterations independently from the explicit depth Elkin & Kurlin (2022a, Definition 3.2).

Assuming that we have already constructed a compressed cover tree on a reference set R, the two main results estimate the time complexity of a new k-nearest neighbor method in Algorithm 4.3m which finds all k-nearest neighbors of any query point $q \in X$ in a reference set $R \subseteq X$ as follows:

- Corollary 4.7 bounds the time complexity as $O\Big(\log_2(k) \cdot (\log_2(\Delta(R)) + |\bar{B}(q, 5d_k(q, R))|)\Big)$, where $\Delta(R)$ is the aspect ratio and $c_m(R)$ is considered fixed (hence hidden).
- Theorem 4.9 bounds the time complexity as $O\Big(\log_2(k) \cdot \big(\log_2(|R|) + k\big)\Big)$, where the expansion constant $c(R \cup \{q\})$ is considered fixed (hence hidden).

Definition 4.1 (λ-point)**.** *Fix a query point q in a metric space (X, d) and fix any level $i \in \mathbb{Z}$. Let $\mathcal{T}(R)$ be its compressed cover tree on a finite reference set $R \subseteq X$. Let C be a subset of a cover set C_i from Definition 2.1 satisfying $\sum\limits_{p \in C} |\mathcal{S}_i(p, \mathcal{T}(R))| \geq k$, where $\mathcal{S}_i(p, \mathcal{T}(R))$ is the distinctive descendant set from Definition 2.8. For any $k \geq 1$, define $\lambda_k(q, C)$ as a point $\lambda \in C$ that minimizes $d(q, \lambda)$ subject to $\sum\limits_{p \in N(q;\lambda)} |\mathcal{S}_i(p, \mathcal{T}(R))| \geq k$.*

Definition 4.2. *Let R be a finite subset of a metric space (X, d). Let $\mathcal{T}(R)$ be a cover tree of Definition 2.1 built on R and let $q \in X$ be arbitrary point. Let $L(\mathcal{T}(R), q)$ be the set of all levels i during iterations of lines 4-15 of Algorithm 4.3 launched with inputs $\mathcal{T}(R), q$. If Algorithm 4.3 reaches line 11 at a level $\varrho \in L(\mathcal{T}(R), q)$, then we say that ϱ is* special. *Set $\eta(i) = \min\{t \in L(\mathcal{T}(R), q) \mid t > i\}$.*

Algorithm 4.3 k-nearest neighbor search by a compressed cover tree

1: **Input** : compressed cover tree $\mathcal{T}(R)$, a query point $q \in X$, an integer $k \in \mathbb{Z}_+$
2: Set $i \leftarrow l_{\max}(\mathcal{T}(R)) - 1$ and $\eta(l_{\max} - 1) = l_{\max}$
3: Let r be the root node of $\mathcal{T}(R)$. Set $R_{l_{\max}} = \{r\}$.
4: **while** $i \geq l_{\min}$ **do**
5: $\quad V = \cup_{q \in R_{\eta(i)}} \{a \in \text{Children}(q) \mid l(a) = i\}$.
6: $\quad$ Assign $\mathcal{C}_i(R_{\eta(i)}) \leftarrow R_{\eta(i)} \cup V$.
7: $\quad$ Compute $\lambda = \lambda_k(q, \mathcal{C}_i(R_{\eta(i)}))$ by Algorithm D.8.
8: $\quad R_i = \{p \in \mathcal{C}_i(R_{\eta(i)}) \mid d(q, p) \leq d(q, \lambda) + 2^{i+2}\}$
9: $\quad$ **if** $d(q, \lambda) > 2^{i+2}$ **then**
10: $\quad\quad$ Collect the distinctive descendants $\mathcal{S}_i(p, \mathcal{T}(R))$ of all points $p \in R$ in set S, see Algorithm F.3.
11: $\quad\quad$ Compute and **output** k-nearest neighbors of the query point q from set S.
12: $\quad$ **end if**
13: $\quad$ Set $j \leftarrow \max_{a \in R_i} \text{Next}(a, i, \mathcal{T}(R))$ $\{$If such j is undefined, we set $j = l_{\min} - 1\}$
14: $\quad$ Set $\eta(j) \leftarrow i$ and $i \leftarrow j$.
15: **end while**
16: Compute and **output** k-nearest neighbors of query point q from the set $R_{l_{\min}}$.

Theorem 4.4 (correctness of Algorithm 4.3)**.** *Algorithm 4.3 correctly finds all k-nearest neighbors of query point q within the reference set R.*

Lemma 4.5. *Algorithm 4.3 has the following time complexities of its lines*

(a) $\max\{\#\text{Line}[4-9], \#\text{Line}[12-15], \#\text{Line}[16]\} = O\big(c_m(R)^{10} \cdot \log_2(k)\big)$*;*

(b) $\#\text{Line}[8-14] = O\big(|\bar{B}(q, 5d_k(q,R))| \cdot \log_2(k)\big)$.

Theorem 4.6. *Let R be a finite set in a metric space (X,d), $c_m(R)$ be the minimized constant from Definition 1.4. Given a compressed cover tree $\mathcal{T}(R)$, Algorithm 4.3 finds all k-nearest neighbors of a query point $q \in X$ in time*

$$O\Big(\log_2(k) \cdot ((c_m(R))^{10} \cdot |L(q, \mathcal{T}(R))| + |\bar{B}(q, 5d_k(q,R))|)\Big),$$

where $L(\mathcal{T}(R), q)$ is the set of all performer iterations (lines 4-15) of Algorithm 4.3.

Proof. Apply Lemma 4.5 to estimate the time complexity of Algorithm 4.3:
$O\big(|L(\mathcal{T}(R), q)| \cdot (\#\text{Line}[4-9] + \#\text{Line}[12-15] + \#\text{Line}[16]) + \#\text{Line}[9-12]\big)$. $\square$

Corollary 4.7 gives a run-time bound using only minimized expansion constant $c_m(R)$, where if $R \subset \mathbb{R}^m$, then $c_m(R) \leq 2^m$. Recall that $\Delta(R)$ is the aspect ratio of R introduced in Definition 1.1.

Corollary 4.7. *Let R be a finite set in a metric space (X,d). Given a compressed cover tree $\mathcal{T}(R)$, Algorithm 4.3 finds all k-nearest neighbors of q in time $O\Big((c_m(R))^{10} \cdot \log_2(k) \cdot \log_2(\Delta(R)) + |\bar{B}(q, 5d_k(q,R))| \cdot \log_2(k)\Big)$.*

Proof. Replace $|L(q, \mathcal{T}(R))|$ in the time complexity of Theorem 4.6 by its upper bound in Lemma 2.7: $|L(q, \mathcal{T}(R))| \leq |H(\mathcal{T}(R))| \leq \log_2(\Delta(R))$. $\square$

Lemma 4.8 is proved similarly to Lemma 3.8. For full details see Appendix G.

Lemma 4.8. *Algorithm 4.3 executes lines 4-15 the following number of times: $|L(\mathcal{T}(R), q)| = O(c(R \cup \{q\})^2 \cdot \log_2(|R|))$.*

Theorem 4.9. *Let R be a finite reference set in a metric space (X,d). Let $q \in X$ be a query point, $c(R \cup \{q\})$ be the expansion constant of $R \cup \{q\}$ and $c_m(R)$ be the minimized expansion constant from Definition 1.4. Given a compressed cover tree $\mathcal{T}(R)$, Algorithm 4.3 finds all k-nearest neighbors of q in time $O\Big(c(R \cup \{q\})^2 \cdot \log_2(k) \cdot ((c_m(R))^{10} \cdot \log_2(|R|) + c(R \cup \{q\}) \cdot k)\Big)$.*

Proof. By Theorem 4.6 the required time complexity is $O\Big((c_m(R))^{10} \cdot \log_2(k) \cdot |L(q, \mathcal{T}(R))| + |\bar{B}(q, 5d(q,\beta))| \cdot \log_2(k)\Big)$, for some point β among the first k-nearest neighbors of q. Apply Definition 1.4 to get the upper bound

$$|B(q, 5d(q,\beta))| \leq (c(R \cup \{q\}))^3 \cdot |B(q, \frac{5}{8}d(q,\beta))| \quad (1)$$

Since $|B(q, \frac{5}{8}d(q,\beta))| \leq k$, we have $|B(q, 5d(q,\beta))| \leq (c(R \cup \{q\}))^3 \cdot k$. It remains to apply Lemma 4.8: $|L(q, \mathcal{T}(R))| = O(c(R \cup \{q\})^2 \cdot \log_2 |R|)$. $\square$

5. Discussion Of Contributions and Next Steps

This paper rigorously proved the time complexity of the exact k-nearest neighbor search. The submission to ICML is strongly motivated by the past gaps in the proofs of time complexities in the highly cited Beygelzimer et al. (2006a, Theorem 5) at ICML, Ram et al. (2009, Theorem 3.1) at NIPS, and March et al. (2010, Theorem 5.1) at KDD.

Though Elkin & Kurlin (2022a) provided concrete counterexamples, no corrections were published. Main Theorem 4.9 and Corollary 3.10 finally filled all the gaps.

Since the past obstacles were caused by unclear descriptions and missed proofs, often without pseudo-codes, this paper necessarily fills in all technical details. Otherwise, future generations would continue citing unreliable results.

To overcome the discovered challenges, first Definition 1.2 and Problem 1.3 rigorously dealt with a potential ambiguity of k-nearest neighbors at equal distances. This singular case was unfortunately not discussed in the past work at all.

A new compressed cover tree in Definition 2.1 substantially simplified the navigating net Krauthgamer & Lee (2004) and original cover tree Beygelzimer et al. (2006a) by avoiding repetitions of given data points. This compression clarified the construction and search in Algorithms 3.4 and 4.3.

Sections 3 and 4 corrected the approach of Beygelzimer et al. (2006a) as follows. Assuming that the expansion constants and aspect ratio of a reference set R are fixed, Corollaries 3.10 and 4.9 rigorously showed that the time complexities are linear in the maximum size of R, Q and near-linear $O(k \log k)$ in the number k of neighbors.

The library MLpack (Curtin et al., 2013) implemented a version of an explicit cover tree, which was later defined in Elkin & Kurlin (2022a, Counterexample 4.2). The implementation of a compressed cover tree is similar but conceptually simpler due to its easier structure in Fig. 1.

The new results justify that the MLpack implementations of the k-nearest neighbors search now have proved theoretical guarantees for a near-linear time complexity, which was practically important for the recent advances below.

Main Theorem 4.9 helped justify a near-linear time complexity for several invariants based on computing k-nearest neighbors in a new area of *Geometric Data Science*, whose aim is to build continuous geographic-style maps for moduli space of real data objects parametrized by complete invariants under practically important equivalence relations.

The key example is a finite cloud of unlabeled points up to isometry maintaining all inter-point distances. The most general isometry invariant SDD (*Simplexwise Distance Distribution* (Kurlin, 2023a)) is conjectured to be complete for any fintie point clouds in any metric space.

In a Euclidean space $\mathbb{R}^n$, the SDD was adapted to the stronger invariant SCD (*Simplexwise Centered Distribution* (Widdowson & Kurlin, 2023)), whose completeness and polynomial complexity (in the number m of points for a fixed dimension n) was proved in (Kurlin, 2023b).

The related and much harder problem is for periodic sets of unlabeled points, which model all solid crystalline materials (periodic crystals). The first generically complete invariant using k-nearest neighbors was the sequence of *density functions* $\psi_k(t)$ measuring a fractional volume of k-fold intersections of balls with a variable radius t and centers at all atoms of a crystal (Edelsbrunner et al., 2021).

These density functions have efficient algorithms in the low dimensions $n = 2, 3$ through higher-degree Voronoi domains (Smith & Kurlin, 2022) of periodic point sets.

The first continuous and complete invariant for periodic point sets in $\mathbb{R}^n$ is the *isoset* of local atomic environments up to a justified stable radius (Anosova & Kurlin, 2021). The first continuous metric on isosets was introduced in (Anosova & Kurlin, 2022) with an approximate algorithm that has a polynomial time complexity (for a fixed dimension n) and a small approximation factor (about 4 in $\mathbb{R}^3$).

The much faster generically complete isometry invariant for both finite and periodic sets of points is the PDD (Pointwise Distance Distribution (Widdowson & Kurlin, 2021)) consisting of distances to k nearest neighbors per point.

The implemented search for atomic neighbors was so fast that all (more than 660 thousand) periodic crystals in the world's largest database of real materials were hierarchically compared by the PDD and its simplified version AMD (Average Minimum Distance (Widdowson et al., 2022)).

Due to the ultra-fast running time, more than 200 billion pairwise comparisons were completed over two days on a modest desktop while past tools were estimated to require over 34 thousand years (Widdowson & Kurlin, 2022).

The most important conclusion from the search results is the *Crystal Isometry Principle* saying that any real periodic crystal has a uniquely defined location in a single continuous space of all isometry classes of periodic point sets (Widdowson & Kurlin, 2022).

This *Crystal Isometry Space* contains all known and not yet discovered crystals similar to the much simpler and discrete Mendeleev's table of chemical elements.

The next step is to improve the complexity of the k-nearest neighbor search to a purely linear time $O(c(R)^{O(1)}|R|)$ with no other extra hidden parameters by using a new compressed cover tree on both sets Q, R.

Since a similar approach Ram et al. (2009) was shown to have incorrect proof in Elkin & Kurlin (2022a, Counterexample 6.5) and Curtin et al. (2015) used some additional parameters I, θ, this goal will require significantly more effort to understand if $O(c(R)^{O(1)}|R|)$ is achievable by using a compressed cover tree.

We thank all reviewers for their time and helpful suggestions. This work was supported by the EPSRC grants EP/R018472/1, EP/X018474/1, and the Royal Academy Engineering fellowship IF2122/186 of the second author.

References

Andoni, A., Indyk, P., and Razenshteyn, I. Approximate nearest neighbor search in high dimensions. In *Proceedings of the International Congress of Mathematicians: Rio de Janeiro 2018*, pp. 3287–3318, 2018.

Anosova, O. and Kurlin, V. An isometry classification of periodic point sets. In *Lecture Notes in Computer Science (Proceedings of DGMM)*, volume 12708, pp. 229–241, 2021.

Anosova, O. and Kurlin, V. Algorithms for continuous metrics on periodic crystals. *arxiv:2205.15298*, 2022.

Arya, S. and Mount, D. M. Approximate nearest neighbor queries in fixed dimensions. In *SODA*, volume 93, pp. 271–280, 1993.

Beckmann, N., Kriegel, H.-P., Schneider, R., and Seeger, B. The r*-tree: An efficient and robust access method for points and rectangles. In *Proceedings of the ACM SIGMOD International Conf. on Management of Data*, pp. 322–331, 1990.

Bentley, J. and Friedman, J. Fast algorithms for constructing minimal spanning trees in coordinate spaces. *IEEE Transactions on Computers*, 27(02):97–105, 1978.

Bentley, J. L. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18 (9):509–517, 1975.

Berchtold, S., Keim, D., and Kriegel, H. The x-tree: An index structure for high-dimensional data. In *Very Large Data-Bases*, pp. 28–39, 1996.

Beygelzimer, A., Kakade, S., and Langford, J. Cover trees for nearest neighbor. In *Proceedings of ICML*, pp. 97–104, 2006a.

Beygelzimer, A., Kakade, S., and Langford, J. Extended version of "cover trees for nearest neighbor". 2006b. URL `https://hunch.net/~jl/projects/cover_tree/paper/paper.pdf`.

Cole, R. and Gottlieb, L.-A. Searching dynamic point sets in spaces with bounded doubling dimension. In *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pp. 574–583, 2006.

Cormen, T. *Introduction to algorithms*. MIT Press McGraw-Hill, Cambridge, Mass. New York, 1990.

Curtin, R. R. *Improving dual-tree algorithms*. PhD thesis, Georgia Institute of Technology, 2015.

Curtin, R. R., Cline, J. R., Slagle, N. P., March, W. B., Ram, P., Mehta, N. A., and Gray, A. G. Mlpack: A scalable c++ machine learning library. *Journal of Machine Learning Research*, 14(Mar):801–805, 2013.

Curtin, R. R., Lee, D., March, W. B., and Ram, P. Plug-and-play dual-tree algorithm runtime analysis. *J. Mach. Learn. Res.*, 16:3269–3297, 2015.

Edelsbrunner, H., Heiss, T., Kurlin, V., Smith, P., and Wintraecken, M. The density fingerprint of a periodic point set. In *Proceedings of Symposium on Computational Geometry*, pp. 32:1–32:16, 2021.

Elkin, Y. and Kurlin, V. Counterexamples expose gaps in the proof of time complexity for cover trees introduced in 2006. In *Topological Data Analysis and Visualization (TopoInVis, arxiv:2208.09447)*, 2022a.

Elkin, Y. and Kurlin, V. Paired compressed cover trees guarantee a near linear parametrized complexity for all k-nearest neighbors search in an arbitrary metric space. *arXiv:2201.06553*, 2022b.

Finkel, R. A. and Bentley, J. L. Quad trees a data structure for retrieval on composite keys. *Acta informatica*, 4(1): 1–9, 1974.

Fukunaga, K. and Narendra, P. M. A branch and bound algorithm for computing k-nearest neighbors. *IEEE transactions on computers*, 100(7):750–753, 1975.

Fussell, D. and Subramanian, K. R. *Fast ray tracing using kd trees*. University of Texas at Austin, Department of Computer Sciences, 1988.

Har-Peled, S. and Mendel, M. Fast construction of nets in low-dimensional metrics and their applications. *SIAM Journal on Computing*, 35(5):1148–1184, 2006.

Holmes, M. P., Gray, A. G., and Isbell Jr, C. L. Quic-svd: Fast svd using cosine trees. In *NIPS*, pp. 673–680, 2008.

Izbicki, M. and Shelton, C. Faster cover trees. In *International Conference on Machine Learning*, pp. 1162–1170. PMLR, 2015.

Jahanseir, M. and Sheehy, D. Transforming hierarchical trees on metric spaces. In *Canadian Conference on Computational Geometry*, pp. 107–113, 2016.

Jones, F. *Lebesgue integration on Euclidean space*. Jones and Bartlett, Sudbury, MA, November 2000.

Kollar, T. *Fast nearest neighbors*. Technical report, Computer Science and Artificial Intelligence Lab, 2006.

Krauthgamer, R. and Lee, J. R. Navigating nets: Simple algorithms for proximity search. In *Proceedings of the fifteenth annual ACM-SIAM symposium on Discrete algorithms*, pp. 798–807, 2004.

Kurlin, V. Simplexwise distance distributions for finite spaces with metrics and measures. *arXiv:2303.14161*, 2023a.

Kurlin, V. The strength of a simplex is the key to a continuous isometry classification of Euclidean clouds of unlabelled points. *arXiv:2303.13486*, 2023b.

Lin, K., Jagadish, H., and Faloutsos, C. The tv-tree: An index structure for high-dimensional data. *The VLDB Journal*, 3(4):517–542, 1994.

Liu, T., Moore, A. W., Gray, A. G., and Yang, K. An investigation of practical approximate nearest neighbor algorithms. In *NIPS*, volume 12, pp. 2004, 2004.

Manocha, S. and Girolami, M. A. An empirical analysis of the probabilistic k-nearest neighbour classifier. *Pattern Recognition Letters*, 28(13):1818–1824, 2007.

March, W. B., Ram, P., and Gray, A. G. Fast euclidean minimum spanning tree: algorithm, analysis, and applications. In *Proceedings of SIG KDD: Knowledge discovery and data mining*, pp. 603–612, 2010.

McNames, J. A fast nearest-neighbor algorithm based on a principal axis search tree. *IEEE Transactions on pattern analysis and machine intelligence*, 23(9):964–976, 2001.

Omohundro, S. M. *Five balltree construction algorithms*. International Computer Science Institute Berkeley, 1989.

Pelleg, D. and Moore, A. Accelerating exact k-means algorithms with geometric reasoning. In *Proceedings of the SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, pp. 277–281, 1999.

Ram, P. and Gray, A. G. Maximum inner-product search using cone trees. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 931–939, 2012.

Ram, P., Lee, D., March, W., and Gray, A. Linear-time algorithms for pairwise statistical problems. *Advances in Neural Information Processing Systems*, 22:1527–1535, 2009.

Ram, P., Lee, D., and Gray, A. G. Nearest-neighbor search on a time budget via max-margin trees. In *Proceedings of the 2012 SIAM International Conference on Data Mining*, pp. 1011–1022. SIAM, 2012.

Rudin, W. *Functional Analysis*. International Series in Pure & Applied Mathematics. McGraw Hill Higher Education, Maidenhead, England, 2 edition, October 1990.

Smith, P. and Kurlin, V. A practical algorithm for degree-k voronoi domains of three-dimensional periodic point sets. In *Lecture Notes in Computer Science (Proceedings of ISVC)*, volume 13599, pp. 377–391, 2022.

Wang, M., Xu, X., Yue, Q., and Wang, Y. A comprehensive survey and experimental comparison of graph-based approximate nearest neighbor search. *arXiv preprint arXiv:2101.12631*, 2021.

Widdowson, D. and Kurlin, V. Pointwise distance distributions of periodic sets. *arXiv:2108.04798*, 2021.

Widdowson, D. and Kurlin, V. Resolving the data ambiguity for periodic crystals. *Advances in Neural Information Processing Systems (Proceedings of NeurIPS 2022)*, 35, 2022.

Widdowson, D. and Kurlin, V. Recognizing rigid patterns of unlabeled point clouds by complete and continuous isometry invariants with no false negatives and no false positives. In *Proceedings of CVPR (arxiv:2303.15385)*, 2023.

Widdowson, D., Mosca, M., Pulido, A., Kurlin, V., and Cooper, A. Average minimum distances of periodic point sets. *MATCH Communications in Mathematical and in Computer Chemistry*, 87:529–559, 2022.

Yianilos, P. N. Data structures and algorithms for nearest neighbor search in general metric spaces. In *Symposium on Discrete Algorithms*, volume 93, pp. 311–21, 1993.

The appendices below contain the full version of the paper with detailed proofs and pseudo codes

A. The k-nearest neighbor search and overview of results

In the modern formulation, k-nearest neighbors problem intends to discover all $k \geq 1$ nearest neighbors in a given reference set R for all points from another given query set Q. Both sets belong to a common ambient space X with a distance d satisfying all metric axioms. The simplest example of X is $\mathbb{R}^n$ with the Euclidean metric. A query set Q can be a single point or a subset of a larger reference set R.

The *exact* k-nearest neighbor problem asks for all true (non-approximate) k-nearest neighbors in R for every query point $q \in Q$. Another probabilistic version of the k-nearest neighbor search Har-Peled & Mendel (2006); Manocha & Girolami (2007) aims to find exact k-nearest neighbors with a given probability. The probabilistic k-nearest neighbor problem can be simplified to k instances of 1-nearest-neighbors problem by splitting R into k subsets $R_1, ..., R_k$ and searching for nearest neighbors in each subset. The approximate version Arya & Mount (1993); Krauthgamer & Lee (2004); Andoni et al. (2018); Wang et al. (2021) of the nearest neighbor search looks for an ϵ-approximate neighbor $r \in R$ of every query point $q \in Q$ such that $d(q, r) \leq (1 + \epsilon)d(q, \mathrm{NN}(q))$, where $\epsilon > 0$ is fixed and $\mathrm{NN}(q)$ is the exact first nearest neighbor of q.

Spacial data structures. It is well known that the time complexity of a brute-force approach of finding all 1st nearest neighbors of points from Q within R is proportional to the product $|Q| \cdot |R|$ of the sizes of Q, R. Already in the 1970s real data was big enough to motivate faster algorithms and sophisticated data structures. One of the first spacial data structures, a *quadtree* Finkel & Bentley (1974), hierarchically splits a reference set $R \subset \mathbb{R}^2$ by subdividing its bounding box (a root) into four smaller boxes (children), which are recursively subdivided until final boxes (leaf nodes) contain only a small number of reference points. A generalization of the quadtree to $\mathbb{R}^n$ exposes an exponential dependence of its computational complexity on the dimension n, because the n-dimensional box is subdivided into 2^n smaller boxes.

The first attempt to overcome this curse of dimensionality was the kd-tree Bentley (1975) that subdivides a subset of R at every step into two subsets instead of 2^n subsets. Many more advanced algorithms utilizing spatial data structures have positively impacted various related research areas such as a minimum spanning tree Bentley & Friedman (1978), range search Pelleg & Moore (1999), k-means clustering Pelleg & Moore (1999), and ray tracing Fussell & Subramanian (1988). The spacial data structures for finding nearest neighbors in the chronological order are k-means tree Fukunaga & Narendra (1975), R tree Beckmann et al. (1990), ball tree Omohundro (1989), R^* tree Beckmann et al. (1990), vantage-point tree Yianilos (1993), TV trees Lin et al. (1994), X trees Berchtold et al. (1996), principal axis tree McNames (2001), spill tree Liu et al. (2004), cover tree Beygelzimer et al. (2006a), cosine tree Holmes et al. (2008), max-margin tree Ram et al. (2012), cone tree Ram & Gray (2012) and others.

Definition 1.1 (diameter and aspect ratio). *For any finite set R with a metric d, the* diameter *is* $\mathrm{diam}(R) = \max\limits_{p \in R} \max\limits_{q \in R} d(p, q)$. *The* aspect ratio *is* $\Delta(R) = \dfrac{\mathrm{diam}(R)}{d_{\min}(R)}$, *where $d_{\min}(R)$ is the shortest distance between points of R.*

Definition 1.2 (k-nearest neighbor set NN_k). *For any point $q \in Q$, let $d_1 \leq \cdots \leq d_{|R|}$ be ordered distances from q to all points of a reference set R whose size (number of points) is denoted by $|R|$. For any $k \geq 1$, the k-nearest neighbor set $\mathrm{NN}_k(q; R)$ consists of all $u \in R$ with $d(q, u) \leq d_k$.*

For $Q = R = \{0, 1, 2, 3\}$, the point $q = 1$ has ordered distances $d_1 = 0 < d_2 = 1 = d_3 < d_4 = 2$. The nearest neighbor sets are $\mathrm{NN}_1(1; R) = \{1\}$, $\mathrm{NN}_2(1; R) = \{0, 1, 2\} = \mathrm{NN}_3(1; R)$, $\mathrm{NN}_4(1; R) = R$. So 0 can be a 2nd neighbor of 1, then 2 becomes a 3rd neighbor of 1, or these neighbors of 0 can be found in a different order.

Problem 1.3 (all k-nearest neighbors search). *Let Q, R be finite subsets of query and reference sets in a metric space (X, d). For any fixed $k \geq 1$, design an algorithm to exactly find k distinct points from $\mathrm{NN}_k(q; R)$ for all $q \in Q$ so that the parametrized worst-case time complexity is near-linear in time* $\max\{|Q|, |R|\}$, *where hidden constants may depend on structures of Q, R but not on their sizes $|Q|, |R|$.*

In a metric space, let $\bar{B}(p, t)$ be the closed ball with a center p and a radius $t \geq 0$. The notation $|\bar{B}(p, t)|$ denotes the number (if finite) of points in the closed ball. Definition 1.4 recalls the expansion constant c from Beygelzimer et al. (2006a) and introduces the new minimized expansion constant c_m, which is a discrete analog of the doubling dimension Cole & Gottlieb (2006).

Definition 1.4 (expansion constants c and c_m). *A subset R of a metric space (X, d) is called* locally finite *if the set $\bar{B}(p, t) \cap R$ is finite for all $p \in X$ and $t \in \mathbb{R}_+$. Let R be a locally finite set in a metric space X.*

The expansion constant $c(R)$ *is the smallest $c(R) \geq 2$ such that $|\bar{B}(p, 2t)| \leq c(R) \cdot |\bar{B}(p, t)|$ for any point $p \in R$ and $t \geq 0$, see Beygelzimer et al. (2006a).*

Introduce the new minimized expansion constant $c_m(R) = \lim\limits_{\xi \to 0^+} \inf\limits_{R \subseteq A \subseteq X} \sup\limits_{p \in A, t > \xi} \dfrac{|\bar{B}(p, 2t) \cap A|}{|\bar{B}(p, t) \cap A|}$, *where A is a locally finite set which covers R.*

Lemma 1.5. *For any finite sets $R \subseteq U$ in a metric space, we have that $c_m(R) \leq c_m(U)$ and $c_m(R) \leq c(R)$.*

Proof. Let us first prove that $c_m(R) \leq c_m(U)$. Let $\epsilon > 0$ be arbitrary real number. By definition of $c_m(U)$ there exists set $\xi > 0$ and set A satisfying $U \subseteq A$ for which

$$\sup_{p \in A, t > \xi} \left| \frac{|\bar{B}(p, 2t) \cap A|}{|\bar{B}(p, t) \cap A|} - c_m(U) \right| \leq \epsilon \tag{2}$$

Since $R \subseteq U$ we have $R \subseteq A$ therefore we can choose the same ξ and set U which satisfy inequality (2). Therefore it follows $c_m(R) \leq c_m(U) + \epsilon$. Since ϵ was chosen arbitrarily it follows that $c_m(R) \leq c_m(U)$.

To prove that $c_m(R) \leq c(R)$, note that $\sup\limits_{p \in A, t > \xi} |\dfrac{|\bar{B}(p, 2t) \cap A|}{|\bar{B}(p, t) \cap A|} \leq \sup\limits_{p \in A, t > 0} |\dfrac{|\bar{B}(p, 2t) \cap A|}{|\bar{B}(p, t) \cap A|}$. Then by choosing $\xi = \frac{d_{\min}(R)}{4}$ and $A = R$ we have:

$$c_m(R) \leq \sup_{p \in R, t > 0} \left| \frac{|\bar{B}(p, 2t) \cap R|}{|\bar{B}(p, t) \cap R|} - c_m(U) \right| = c(R)$$

□

Note that both $c(R), c_m(R)$ are always defined when R is finite. We will show that a single outlier can make the expansion constant $c(R)$ as large as $O(|R|)$. The set $R = \{1, 2, \ldots, n, 2n + 1\}$ of $|R| = n + 1$ points has $c(R) = n + 1$ because $\bar{B}(2n + 1; n) = \{2n + 1\}$ is a single point, while $\bar{B}(2n + 1; 2n) = R$ is the full set of $n + 1$ points. On the other hand the same set R can be extended to a larger uniform set $A = \{1, 2, \ldots, 2n - 1, 2n\}$ whose expansion constant $c(A) = 2$, therefore the minimized constant of the original set R becomes much smaller: $c_m(R) \leq c(A) = 2 < c(R) = n + 1$.

The constant c from Beygelzimer et al. (2006a) equals to $2^{\dim_{KR}}$ from Krauthgamer & Lee (2004, Section 2.1). In Krauthgamer & Lee (2004, Section 1.1) the doubling dimension $2^{\dim}$ is defined as a minimum value ρ such that any set X can be covered by 2^{ρ} sets whose diameters are half of the diameter of X. The past work Krauthgamer & Lee (2004) proves that $2^{\dim} \leq 2^n$ for any subset of $\mathbb{R}^n$. Theorem C.15 will prove that $c_m(R) \leq 2^n$ for any a finite subset $R \subset \mathbb{R}^n$, so $c_m(R)$ mimics $2^{\dim}$.

Navigating nets. In 2004, Krauthgamer & Lee (2004, Theorem 2.7) claimed that a navigating net can be constructed in time $O\big(2^{O(\dim_{KR}(R)}|R|(\log|R|)\log(\log|R|)\big)$ and all k-nearest neighbors of a query point q can be found in time $O(2^{O(\dim_{KR}(R \cup \{q\})}(k + \log|R|)$, where $\dim_{KR}(R \cup \{q\})$ is the expansion constant defined above. All proofs and pseudo-codes were omitted. The authors didn't reply to our request for details.

Modified navigating nets Cole & Gottlieb (2006) were used in 2006 to claim the time $O(\log(n) + (1/\epsilon)^{O(1)})$ for the $(1 + \epsilon)$-approximate neighbors. All proofs and pseudo-codes were left out, also for the construction of the modified navigating net for the claimed time $O(|R| \cdot \log(|R|))$.

Cover trees. In 2006, Beygelzimer et al. (2006a) introduced a cover tree inspired by the navigating nets Krauthgamer & Lee (2004). This cover tree was designed to prove a worst-case bound for the nearest neighbor search in terms of the size $|R|$ of a reference set R and the expansion constant $c(R)$ of Definition 1.4. Assume that a cover tree is already constructed on set R. Then Beygelzimer et al. (2006a, Theorem 5) claims that nearest neighbor of any query point $q \in Q$ could be found in time $O(c(R)^{12} \cdot \log|R|)$. In 2015, Curtin (2015, Section 5.3) pointed out that the proof of Beygelzimer et al. (2006a, Theorem 5)

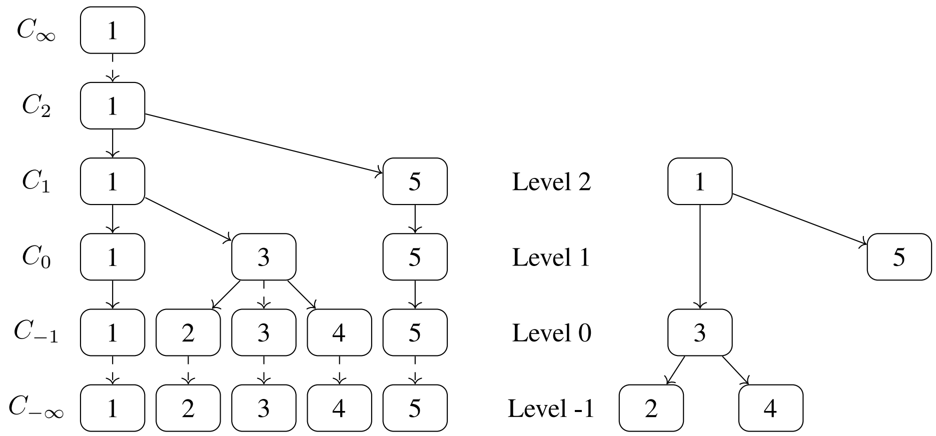

Figure 3. **Left:** an implicit cover tree from Beygelzimer et al. (2006a, Section 2) at ICML 2006 for a finite set of reference points $R = \{1, 2, 3, 4, 5\}$ with the Euclidean distance $d(x, y) = |x - y|$. **Right:** a new compressed cover tree in Definition 2.1 corrects the past worst-case complexity for k-nearest neighbors search in R.

contains a crucial gap, now have been confirmed by a specific dataset in Elkin & Kurlin (2022a, Counterexample 5.2). The time complexity result of the cover tree construction algorithm Beygelzimer et al. (2006a, Theorem 6) had a similar issue, the gap of which is exposed rigorously in Elkin & Kurlin (2022a, Counterexample 4.2).

Further studies in cover trees. A noteworthy paper on cover trees Kollar (2006) introduced a new probabilistic algorithm for the nearest neighbor search, as well as corrected the pseudo-code of the cover tree construction algorithm of Beygelzimer et al. (2006a, Algorithm 2). Later in 2015, a new, more efficient implementation of cover tree was introduced in Izbicki & Shelton (2015). However, no new time-complexity results were proven. A study Jahanseir & Sheehy (2016) explored connections between modified navigating nets Cole & Gottlieb (2006) and cover trees Beygelzimer et al. (2006a). Multiple papers Beygelzimer et al. (2006b); Ram et al. (2009); Curtin et al. (2015) studied possibility of solving k-nearest neighbor problem (Problem 1.3) by using cover tree on both, the query set and the reference set, for further details see Elkin & Kurlin (2022a, Section 6).

The main contributions are the following.

- Definition 2.1 introduces a compressed cover tree.
- Theorem 3.6 and Corollary 3.10 estimate the time to build a compressed cover tree.
- Theorem 4.9 and Corollary 4.7 estimate the time to find all k-nearest neighbors as in Problem 1.3.
- Theorem G.6 estimates the time complexity of approximate k-nearest neighbor search.

This work corrects the past gaps of the single-tree approach via an original cover tree Beygelzimer et al. (2006a) by using a new compressed cover tree $\mathcal{T}(R)$ from Definition 2.1, which can be constructed on any finite reference set R with a metric d. Theorem 3.9 will prove that a compressed cover tree $\mathcal{T}(R)$ can be built in time $O(c_m(R)^8 \cdot c(R)^2 \cdot \log_2(|R|) \cdot |R|)$.

The past gap in the proof of the time complexity Beygelzimer et al. (2006a, Theorem 1) for nearest neighbor search is tackled by new Algorithm F.2, which add an essential block to the original code in Beygelzimer et al. (2006a, Algorithm 1). The extra block eliminates the issue of having too many successive iterations when a query point q is disproportionately far away from the remaining candidate set R_i on some level i. Then Lemma 4.8 shows that the number of iterations of Algorithm F.2 is bounded by $O(c(R)^2 \log_2(|R|))$. This new lemma replaces the old result Beygelzimer et al. (2006a, Lemma 4.3), which had a similar bound for the number of explicit levels of a cover tree, for further information see Elkin & Kurlin (2022a, Definition 3.2) The old result cannot be used to estimate the number of iterations of Beygelzimer et al. (2006a, Algorithm 1) due to Elkin & Kurlin (2022a, Counterexample 5.2).

Assume that a compressed cover tree $\mathcal{T}(R)$ is already constructed on a reference set R. Our first main Theorem 4.9 shows that k-nearest neighbors of a query node q can be found in time of

$$O\Big(c(R \cup \{q\})^2 \cdot \log_2(k) \cdot \big((c_m(R))^{10} \cdot \log_2(|R|) + c(R \cup \{q\}) \cdot k\big)\Big).$$

Recall that $c(R)$ can potentially become as large as $O(|R|)$ when R is not uniformly distributed. Our second main Corollary 4.7 estimates the time complexity of the new k-nearest neighbor search by using only the minimized expansion constant $c_m(R)$ of Definition 1.4 and the aspect ratio $\Delta(R)$ of Definition 1.1 as parameters. These parameters are less dependent on the point distribution (or noise) in the sets R, Q. In many cases, $\Delta(R)$ is relatively small and $c_m(R)$ depends mostly on the dimension of the ambient space X. It is shown that k-nearest neighbors of q in a reference set R can be found in time

$$O\Big((c_m(R))^{10} \cdot \log_2(k) \cdot \log_2(\Delta(R)) + |\bar{B}(q, 5d_k(q, R))| \cdot \log_2(k)\Big), \text{ where}$$

$d_k(q, R)$ is the distance from q to its kth nearest neighbor. Tables 7-8 summarize past and new results.

Table 5. Results for building data structures with hidden classic expansion constant $c(R)$ of Definition 1.4 or KR-type constant $2^{\dim_{KR}}$ Krauthgamer & Lee (2004, Section 2.1)

Data structure	claimed time complexity	space	proofs
Navigating nets Krauthgamer & Lee (2004)	$O\big(2^{O(\dim_{KR})} \cdot \|R\| \log(\|R\|) \log(\log \|R\|)\big)$, Krauthgamer & Lee (2004, Theorem 2.6)	$O(2^{O(\dim)}\|R\|)$	Not available
Cover tree Beygelzimer et al. (2006a)	$O(c(R)^{O(1)} \cdot \|R\| \cdot \log \|R\|)$, Beygelzimer et al. (2006a, Theorem 6)	$O(\|R\|)$	Elkin & Kurlin (2022a, Counterexample 4.2) shows that the past proof is incorrect
Compressed cover tree [dfn 2.1]	$O\big(c(R)^{O(1)} \cdot \|R\| \cdot \log(R)\big)$	$O(\|R\|)$ Lemma B.1	Corollary 3.10

Table 6. Results for exact k-nearest neighbors of one query point $q \in X$ using hidden classic expansion constant $c(R)$ of Definition 1.4 or KR-type constant $2^{\dim_{KR}}$ Krauthgamer & Lee (2004, Section 2.1) and assuming that all data structures are already built. Note that the dimensionality factor $2^{\dim_{KR}}$ is equivalent to $c(R)^{O(1)}$.

Data structure	claimed time complexity	space	proofs
Navigating nets Krauthgamer & Lee (2004)	$O\big(2^{O(\dim_{KR})}(\log(\|R\|) + k)\big)$ for $k \geq 1$ Krauthgamer & Lee (2004, Theorem 2.7)	$O(2^{O(\dim)}\|R\|)$	Not available
Cover tree Beygelzimer et al. (2006a)	$O\big(c(R)^{O(1)} \log \|R\|\big)$ for $k = 1$ Beygelzimer et al. (2006a, Theorem 5)	$O(\|R\|)$	Elkin & Kurlin (2022a, Counterexample 5.2) shows that the past proof is incorrect
Compressed cover tree, Definition 2.1	$O\big(c(R)^{O(1)} \cdot \log(k) \cdot (\log(\|R\|) + k)\big)$	$O(\|R\|)$, Lemma B.1	Theorem 4.9

Table 7. Building data structures with hidden $c_m(R)$ or dimensionality constant 2^{dim} Krauthgamer & Lee (2004, Section 1.1)

Data structure	claimed time complexity	space	proofs
Navigating nets Krauthgamer & Lee (2004)	$O\big(2^{O(\text{dim})} \cdot \lvert R\rvert \cdot \log(\Delta) \cdot \log(\log((\Delta))\big)$	$O(2^{O(\text{dim})}\lvert R\rvert)$	Krauthgamer & Lee (2004, Theorem 2.5)
Compressed cover tree [dfn 2.1]	$O\big(c_m(R)^{O(1)} \cdot \lvert R\rvert \log(\Delta(\lvert R\rvert))\big)$	$O(\lvert R\rvert)$ Lemma B.1	Theorem 3.6

Table 8. Results for exact k-nearest neighbors of one point q using hidden $c_m(R)$ or dimensionality constant 2^{dim} Krauthgamer & Lee (2004, Section 1.1) assuming that all structures are built.

Data structure	claimed time complexity	space	proofs
Navigating nets Krauthgamer & Lee (2004)	$O\big(2^{O(\text{dim})} \cdot \log(\Delta) + \lvert\bar{B}(q, O(d(q,R))\rvert\big)$ for $k=1$	$O(2^{O(\text{dim})}\lvert R\rvert)$	a proof outline in Krauthgamer & Lee (2004, Theorem 2.3)
Compressed cover tree, Definition 2.1	$O\big(c_m(R)^{O(1)} \cdot \log(k) \cdot (\log(\lvert\Delta\rvert) + \lvert\bar{B}(q, O(d_k(q,R))\rvert)\big)$	$O(\lvert R\rvert)$, Lemma B.1	Corollary 4.7

B. Compressed cover tree

This section introduces in Definition 2.1 a new compressed cover tree, which will be used to solve Problem 1.3. Other important results are Lemmas 2.2 and 2.4. Given a δ-sparse finite metric space R, Lemma 2.2 shows that the number of points of R in the closed ball $\bar{B}(p,t)$ has the upper bound $c_m(S)^\mu$, where μ depends on $\frac{t}{\delta}$. Lemma 2.4 will imply that if there are points p, q in a finite metric space R satisfying $2r < d(p,q) \leq 3r$ for some $r \in \mathbb{R}$, then $|\bar{B}(q,4r)| \geq (1 + \frac{1}{c(R)^2})|\bar{B}(q,r)|$.

Definition 2.1 (a compressed cover tree $\mathcal{T}(R)$)**.** *Let R be a finite set in a metric space (X,d).* A compressed cover tree *$\mathcal{T}(R)$ has the vertex set R with a root $r \in R$ and a* level *function $l : R \to \mathbb{Z}$ satisfying the conditions below.*

(2.1a) Root condition *: the level of the root node r satisfies $l(r) \geq 1 + \max\limits_{p \in R\setminus\{r\}} l(p)$.*

(2.1b) Cover condition *: for every node $q \in R \setminus \{r\}$, we select a unique* parent *p and a level $l(q)$ such that $d(q,p) \leq 2^{l(q)+1}$ and $l(q) < l(p)$; this parent node p has a single link to its* child *node q.*

(2.1c) Separation condition *: for $i \in \mathbb{Z}$, the* cover set *$C_i = \{p \in R \mid l(p) \geq i\}$ has $d_{\min}(C_i) = \min\limits_{p \in C_i} \min\limits_{q \in C_i \setminus \{p\}} d(p,q) > 2^i$.*

Since there is a 1-1 map between R and all nodes of $\mathcal{T}(R)$, the same notation p can refer to a point in the set R or to a node of the tree $\mathcal{T}(R)$. Set $l_{\max} = 1 + \max\limits_{p \in R\setminus\{r\}} l(p)$ and $l_{\min} = \min\limits_{p \in R} l(p)$. For any node $p \in \mathcal{T}(R)$, $\text{Children}(p)$ *denotes the set of all children of p, including p itself. For any node $p \in \mathcal{T}(R)$, define the* node-to-root *path as a unique sequence of nodes $w_0, \dots, w_m$ such that $w_0 = p$, w_m is the root and w_{j+1} is the parent of w_j for $j = 0, ..., m-1$.*

A node $q \in \mathcal{T}(R)$ is a descendant *of a node p if p is in the node-to-root path of q. A node p is an* ancestor *of q if q is in the node-to-root path of p. Let* $\text{Descendants}(p)$ *be the set of all descendants of p, including itself p.*

Lemma B.1 (Linear space of $\mathcal{T}(R)$)**.** *Let (R,d) be a finite metric space. Then any cover tree $\mathcal{T}(R)$ from Definition 2.1 takes $O(|R|)$ space.*

Proof. Since $\mathcal{T}(R)$ is a tree , both its vertex set and its edge set contain at most $|R|$ nodes. Therefore $\mathcal{T}(R)$ takes at most $O(|R|)$ space. ☐

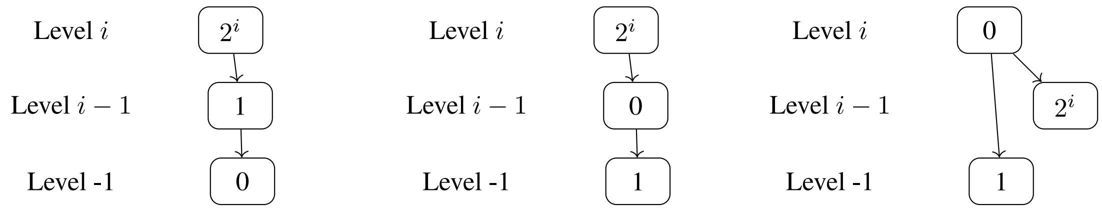

Figure 4. Compressed cover trees $\mathcal{T}(R)$ from Definition 2.1 for $R = \{0, 1, 2^i\}$.

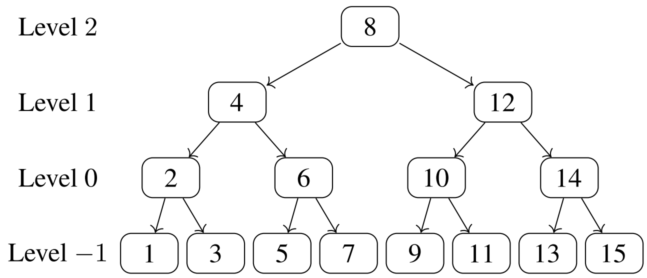

Figure 5. Compressed cover tree $\mathcal{T}(R)$ on the set R in Example B.2 with root 16.

Example B.2 ($\mathcal{T}(R)$ in Fig. 5)**.** Let $(\mathbb{R}, d = |x - y|)$ be the real line with euclidean metric. Let $R = \{1, 2, 3, ..., 15\}$ be its finite subset. Fig. 5 shows a compressed cover tree on the set R with the root $r = 8$. The cover sets of $\mathcal{T}(R)$ are $C_{-1} = \{1, 2, 3, ..., 15\}$, $C_0 = \{2, 4, 6, 8, 10, 12, 14\}$, $C_1 = \{4, 8, 12\}$ and $C_2 = \{8\}$. We check the conditions of Definition 2.1.

- Root condition $(2.1a)$: since $\max_{p \in R \setminus \{8\}} d(p, 8) = 7$ and $\lceil \log_2(7) \rceil - 1 = 2$, the root can have the level $l(8) = 2$.
- Covering condition (2.1b) : for any $i \in -1, 0, 1, 2$, let p_i be arbitrary point having $l(p_i) = i$. Then we have $d(p_{-1}, p_0) = 1 \leq 2^0$, $d(p_0, p_1) = 2 \leq 2^1$ and $d(p_1, p_2) = 4 \leq 2^2$.
- Separation condition (2.1c) : $d_{\min}(C_{-1}) = 1 > \frac{1}{2} = 2^{-1}$, $d_{\min}(C_0) = 2 > 1 = 2^0$, $d_{\min}(C_1) = 4 > 2 = 2^1$. ■

A cover tree was defined in Beygelzimer et al. (2006a, Section 2) as a tree version of a navigating net from Krauthgamer & Lee (2004, Section 2.1). For any index $i \in \mathbb{Z} \cup \{\pm\infty\}$, the level i set of this cover tree coincides with the cover set C_i above, which can have nodes at different levels in Definition 2.1. Any point $p \in C_i$ has a single parent in the set C_{i+1}, which satisfied conditions (2.1b,c). Beygelzimer et al. (2006a, Section 2) referred to this original tree as an implicit representation of a cover tree. Such a tree in Figure 6 (left) contains infinitely many repetitions of every point $p \in R$ in long branches and will be called an *implicit cover tree*.

Since an implicit cover tree is formally infinite, for practical implementations, the authors of Beygelzimer et al. (2006a) had to use another version that they named an explicit representation of a cover tree. We call this version an *explicit cover tree*. Here is the full defining quote at the end of Beygelzimer et al. (2006a, Section 2): "The explicit representation of the tree coalesces all nodes in which the only child is a self-child". In an explicit cover tree, if a subpath of every node-to-root path consists of all identical nodes without other children, all these identical nodes collapse to a single node, see Figure 6 (middle).

Since an explicit cover tree still contains repeated points, Definition 2.1 is well-motivated by the aim to include every point only once, which saves memory and simplifies all subsequent algorithms, see Fig. 6 (right).

Example B.3 (a short train line tree)**.** Let G be the unoriented metric graph consisting of two vertices r, q connected by three different edges e, h, g of lengths $|e| = 2^6$, $|h| = 2^3$, $|g| = 1$. Let p_4 be the middle point of the edge e. Let p_3 be the middle point of the subedge (p_4, q). Let p_2 be the middle point of the edge h. Let p_1 be the middle point of the subedge (p_2, q). Let $R = \{p_1, p_2, p_3, p_4, r\}$. We construct a compressed cover tree $\mathcal{T}(R)$ by choosing the level $l(p_i) = i$ and by setting the root r to be the parent of both p_2 and p_4, p_4 to be the parent of p_3, and p_2 to be the parent of p_1. Then $\mathcal{T}(R)$ satisfies all the conditions of Definition 2.1, see a comparison of the three cover trees in Fig. 6. ■

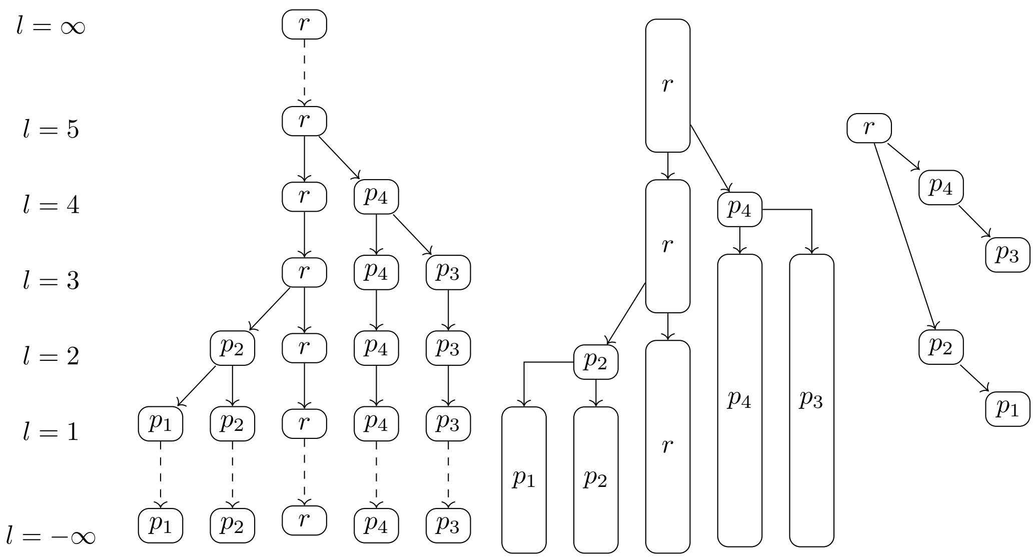

Figure 6. A comparison of past cover trees and a new tree in Example B.3. **Left:** an implicit cover tree contains infinite repetitions. **Middle:** an explicit cover tree. **Right:** a compressed cover tree from Definition 2.1 includes each point once.

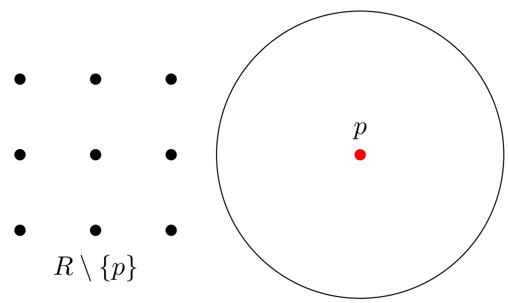

Figure 7. Example B.4 describes a set R with a big expansion constant $c(R)$. Let $R \setminus \{p\}$ be a finite subset of a unit square lattice in $\mathbb{R}^2$, but a point p is located far away from $R \setminus \{p\}$ at a distance larger than $\mathrm{diam}(R \setminus \{p\})$. Definition 1.4 implies that $c(R) = |R|$.

Even a single outlier point can make the expansion constant big. Consider set $R = \{1, 2, ..., n-1, 2n\}$ for some $n \in \mathbb{Z}_+$. Since $|\bar{B}(2n, n)| = 1$ and $|\bar{B}(2n, n)| = |R|$, we have $c(R) = |R|$. Example B.4 shows that expansion constant of a set R can be as big as $|R|$.

Example B.4 (one outlier can make the expansion constant big)**.** Let R be a finite metric space and $p \in R$ satisfy $d(p, R \setminus \{t\}) > \mathrm{diam}(R \setminus \{p\})$. Since $\bar{B}(p, 2d(p, R \setminus \{t\}) = R$, $\bar{B}(p, d(p, R \setminus \{t\}) = \{p\}$, we get $c(R) = N$, see Fig. 7. ■

Example B.5 shows that the minimized expansion can be significantly smaller than the original expansion constant.

Example B.5 (minimized expansion constants)**.** Let $(\mathbb{R}, d)$ be the Euclidean line. For an integer $n > 10$, consider the finite sets $R = \{2^i \mid i \in [1, n]\}$ and let $Q = \{i \mid i \in [1, 2^n]\}$. If $0 < \epsilon < 10^{-9}$, then $\bar{B}(2^n, 2^{n-1} - \epsilon) = \{2^n\}$ and $\bar{B}(2^n, 2(2^{n-1} - \epsilon)) = R$, so $c(R) = n$. For any $q \in Q$ and any $t \in \mathbb{R}$, we have that $\bar{B}(q, t) = \mathbb{Z} \cap [q - t, q + t]$ and $\bar{B}(q, 2t) = \mathbb{Z} \cap [q - 2t, q + 2t]$, so $c(Q) \leq 4$. Then $c_m(R) \leq c_m(Q) \leq c(Q) \leq 4$ by Lemma 1.5. ■

Lemma B.6 provides an upper bound for a distance between a node and its descendants.

Lemma B.6 (a distance bound on descendants)**.** *Let R be a finite subset of an ambient space X with a metric d. In a compressed cover tree $\mathcal{T}(R)$, let q be any descendant of a node p. Let the node-to-root path S of q contain a node u satisfying $u \in \mathrm{Children}(p) \setminus \{p\}$. Then $d(p, q) \leq 2^{l(u)+2} \leq 2^{l(p)+1}$.* ■

Proof. Let $(w_0, ..., w_m)$ be a subpath of the node-to-root path for $w_0 = q$, $w_{m-1} = u$, $w_m = p$. Then $d(w_i, w_{i+1}) \leq$

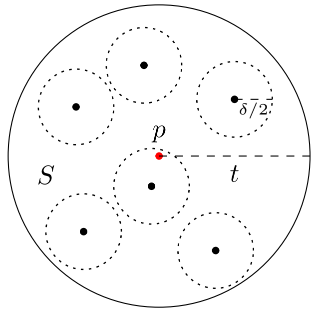

Figure 8. This volume argument proves Lemma 2.2. By using an expansion constant, we can find an upper bound for the number of smaller balls of radius $\frac{\delta}{2}$ that can fit inside a larger $\bar{B}(p,t)$.

$2^{l(w_i)+1}$ for any i. The first required inequality follows from the triangle inequality below:

$$d(p,q) \leq \sum_{j=0}^{m-1} d(w_j, w_{j+1}) \leq \sum_{j=0}^{m-1} 2^{l(w_j)+1} \leq \sum_{t=l_{\min}}^{l(u)+1} 2^t \leq 2^{l(u)+2}$$

Finally, $l(u) \leq l(p)-1$ implies that $d(p,q) \leq 2^{l(p)+1}$. □

Lemma 2.2 uses the idea of Curtin et al. (2015, Lemma 1) to show that if S is a δ-sparse subset of a metric space X, then S has at most $(c_m(S))^{\mu}$ points in the ball $\bar{B}(p,r)$, where $c_m(S)$ is the minimized expansion constant of S, while μ depends on δ, r.

Lemma 2.2 (packing)**.** *Let S be a finite δ-sparse set in a metric space (X,d), so $d(a,b) > \delta$ for all $a,b \in S$. Then, for any point $p \in X$ and any radius $t > \delta$, we have $|\bar{B}(p,t) \cap S| \leq (c_m(S))^{\mu}$, where $\mu = \lceil \log_2(\frac{4t}{\delta}+1) \rceil$.*

Proof. Assume that $d(p,q) > t$ for any point $q \in S$. Then $\bar{B}(p,t) \cap S = \emptyset$ and the lemma holds trivially. Otherwise $\bar{B}(p,t) \cap S$ is non-empty. By Definition 1.4 of a minimized expansion constant, for any small enough $\epsilon > 0$, we can always find $\xi \leq \frac{2t+\frac{\delta}{2}}{2^{\mu}}$ and a set A such that $S \subseteq A \subseteq X$ for which:

$$|B(q,2s) \cap A| \leq (c_m(S)+\epsilon) \cdot |B(q,s) \cap A| \tag{3}$$

for any $q \in A$ and $s > \xi$. Note that for any $u \in \bar{B}(p,t) \cap S$ we have $\bar{B}(u,\frac{\delta}{2}) \subseteq \bar{B}(p,t+\frac{\delta}{2})$. Therefore, for any point $q \in \bar{B}(p,t) \cap S$, we get

$$\bigcup_{u \in \bar{B}(p,t) \cap S} \bar{B}(u,\frac{\delta}{2}) \subseteq \bar{B}(p,t+\frac{\delta}{2}) \subseteq \bar{B}(q,2t+\frac{\delta}{2})$$

Since all the points of S were separated by δ, we have

$$|\bar{B}(p,t) \cap S| \cdot \min_{u \in \bar{B}(p,t) \cap S} |\bar{B}(u,\frac{\delta}{2}) \cap A| \leq \sum_{u \in \bar{B}(p,t) \cap S} |\bar{B}(u,\frac{\delta}{2}) \cap A| \leq |\bar{B}(q,2t+\frac{\delta}{2}) \cap A|$$

In particular, by setting $q = \operatorname{argmin}_{a \in S \cap \bar{B}(p,t)} |\bar{B}(a,\frac{\delta}{2})|$, we get:

$$|\bar{B}(p,t) \cap S| \cdot |\bar{B}(q,\frac{\delta}{2}) \cap A| \leq |\bar{B}(q,2t+\frac{\delta}{2}) \cap A| \tag{4}$$

Inequality (3) applied μ times for the radii $s_i = \dfrac{2t+\frac{\delta}{2}}{2^i}$, $i = 1,...,\mu$, implies that:

$$|\bar{B}(q,2t+\frac{\delta}{2}) \cap A| \leq (c_m(S)+\epsilon)^{\mu} |\bar{B}(q,\frac{2t+\frac{\delta}{2}}{2^{\mu}}) \cap A| \leq (c_m(S)+\epsilon)^{\mu} |\bar{B}(q,\frac{\delta}{2}) \cap A|. \tag{5}$$

By combining inequalities (4) and (5), we get

$$|\bar{B}(p,t) \cap S| \leq \frac{|\bar{B}(q, 2t + \frac{\delta}{2}) \cap A|}{|\bar{B}(q, \frac{\delta}{2}) \cap A|} \leq (c_m(S) + \epsilon)^{\mu}.$$

The required inequality is obtained by letting $\epsilon \to 0$. □

Krauthgamer & Lee (2004, Section 1.1) defined dim(X) of a space (X, d) as the minimum number m such that every set $U \subseteq X$ can be covered by 2^m sets whose diameter is a half of the diameter of U. If U is finite, an easy application of Lemma 2.2 for $\delta = \frac{r}{2}$ shows that $\dim(X) \leq \sup_{A \subseteq X}(c_m(A))^4 \leq \sup_{A \subseteq X} \inf_{A \subseteq B \subseteq X}(c(B))^4$, where A and B are finite subsets of X.

Let $T(R)$ be an implicit cover tree of Beygelzimer et al. (2006a) on a finite set R. Beygelzimer et al. (2006a, Lemma 4.1) showed that the number of children of any node $p \in T(R)$ has the upper bound $(c(R))^4$. Lemma 2.3 generalizes Beygelzimer et al. (2006a, Lemma 4.1) for a compressed cover tree.

Lemma 2.3 (width bound). *Let R be a finite subset of a metric space (X, d). For any compressed cover tree $\mathcal{T}(R)$, any node p and any level $i \leq l(p)$ we have*

$$\{q \in \text{Children}(p) \mid l(q) = i\} \cup \{p\} \leq (c_m(R))^4,$$

where $c_m(R)$ is the minimized expansion constant of R.

Proof. By the covering condition of $\mathcal{T}(R)$, any child q of p located on the level i has $d(q,p) \leq 2^{i+1}$. Then the number of children of the node p at level i at most $|\bar{B}(p, 2^{i+1})|$. The separation condition in Definition 2.1 implies that the set C_i is a 2^i-sparse subset of X. We apply Lemma 2.2 for $t = 2^{i+1}$ and $\delta = 2^i$. Since $4 \cdot \frac{t}{\delta} + 1 \leq 4 \cdot 2 + 1 \leq 2^4$, we get $|\bar{B}(q, 2^{i+1}) \cap C_i| \leq (c_m(C_i))^4$. Lemma 1.5 implies that $(c_m(C_i))^4 \leq (c_m(R))^4$, so the upper bound is proved. □

Lemma 2.4 (growth bound). *Let (A, d) be a finite metric space, let $q \in A$ be an arbitrary point and let $r \in \mathbb{R}$ be a real number. Let $c(A)$ be the expansion constant from Definition 1.4. If there exists a point $p \in A$ such that $2r < d(p,q) \leq 3r$, then $|\bar{B}(q, 4r)| \geq (1 + \frac{1}{c(A)^2}) \cdot |\bar{B}(q, r)|$.*

Proof. Since $\bar{B}(q,r) \subset \bar{B}(p, 3r + r)$, we have $|\bar{B}(q,r)| \leq |\bar{B}(q, 4r)| \leq c(A)^2 \cdot |\bar{B}(p,r)|$. And since $\bar{B}(p,r)$ and $\bar{B}(q,r)$ are disjoint and are subsets of $\bar{B}(q, 4r)$, we have

$$|\bar{B}(q, 4r)| \geq |\bar{B}(q,r)| + |\bar{B}(p,r)| \geq |\bar{B}(q,r)| + \frac{|\bar{B}(q,r)|}{c(A)^2} \geq (1 + \frac{1}{c(A)^2}) \cdot |\bar{B}(q,r)|,$$

which proves the claim. □

Lemma 2.5 (extended growth bound). *Let (A, d) be a finite metric space, let $q \in A$ be an arbitrary point. Let $p_1, ..., p_n$ be a sequence of distinct points in R, in such a way that for all $i \in \{2, ..., n\}$ we have $4 \cdot d(p_i, q) \leq d(p_{i+1}, q)$. Then*

$$|\bar{B}(q, \frac{4}{3} \cdot d(q, p_n))| \geq (1 + \frac{1}{c(A)^2})^n \cdot |\bar{B}(q, \frac{1}{3} \cdot d(q, p_1))|.$$

Proof. Let us prove this by induction. In basecase $n = 1$ define $r = \frac{d(q, p_m)}{3}$. Now by Lemma 2.4 we have

$$|\bar{B}(q, \frac{4}{3} d(q, p_1))| = |\bar{B}(q, 4r)| \geq (1 + \frac{1}{c(A)^2}) \cdot |\bar{B}(q, r)| = |\bar{B}(q, \frac{1}{3} d(q, p_1))|.$$

Assume now that the claim holds for some $i = m$ and let $p_1, ..., p_{m+1}$ be a sequence satisfying the condition of Lemma 2.5. By induction assumption we have $|\bar{B}(q, \frac{4}{3} d(q, p_m))| \geq (1 + \frac{1}{c(A)^2})^m \cdot |\bar{B}(q, \frac{1}{3} d(q, p_1))|$. Let us pick $r = \frac{d(q, p_{m+1})}{3}$. Then we have:

$$\begin{aligned}
|\bar{B}(q, \frac{4}{3} \cdot d(q, p_{m+1}))| &\geq (1 + \frac{1}{c(A)^2}) \cdot |\bar{B}(q, \frac{1}{3} \cdot d(q, p_{m+1}))| \\
&\geq (1 + \frac{1}{c(A)^2}) \cdot |\bar{B}(q, \frac{4}{3} \cdot d(q, p_m))| \\
&\geq (1 + \frac{1}{c(A)^2}) \cdot (1 + \frac{1}{c(A)^2})^m \cdot |\bar{B}(q, \frac{1}{3} \cdot d(q, p_1))| \\
&\geq (1 + \frac{1}{c(A)^2})^{m+1} \cdot |\bar{B}(q, \frac{1}{3} \cdot d(q, p_1))|
\end{aligned}$$

which proves the claim. □

Lemma B.7. *For every $x \in \mathbb{R}$ satisfying $x \geq 2$, the following inequality holds:*

$$x^2 \geq \frac{1}{\log_2(1 + \frac{1}{x^2})}.$$

Proof. Let $\ln$ be natural logarithm. Note first that for any $u > 0$ we have:

$$\frac{u}{u+1} = \int_0^u \frac{dt}{u+1} \leq \int_0^u \frac{dt}{t+1} = \ln(u+1).$$

By setting $u = \frac{1}{x^2} > 0$ we get:

$$\log_2(1 + \frac{1}{x^2}) = \frac{\ln(\frac{1}{x^2})}{\ln(2)} \geq \frac{1}{\ln(2)} \cdot \frac{\frac{1}{x^2}}{\frac{1}{x^2} + 1} = \frac{1}{\ln(2)} \cdot \frac{1}{x^2 + 1}.$$

Let us now show that for $x \geq 2$ we have: $\frac{1}{\ln(2)} \cdot \frac{1}{x^2+1} \geq \frac{1}{x^2}$. Note first that $4 \geq \frac{\ln(2)}{1-\ln(2)}$. Since $x \geq 2$ we have $x^2 \geq \frac{\ln(2)}{1-\ln(2)}$. Therefore $x^2 - \ln(2) \cdot x^2 \geq \ln(2)$ and $x^2 \geq \ln(2) \cdot (1 + x^2)$. It follows that $\frac{1}{\ln(2)} \frac{1}{1+x^2} \geq \frac{1}{x^2}$, which proves the claim.

□

Definition 2.6 (the height of a compressed cover tree). *For a compressed cover tree $\mathcal{T}(R)$ on a finite set R, the* height set *is $H(\mathcal{T}(R)) = \{i \mid C_{i-1} \neq C_i\} \cup \{l_{\max}, l_{\min}\}$. The size $|H(\mathcal{T}(R))|$ of this set is called the* height *of $\mathcal{T}(R)$.*

The new concept of the height $|H(\mathcal{T})|$ will justify a near-linear parameterized worst-case complexity in Theorem 4.9. By condition (2.1b), the height $|H(\mathcal{T}(R))|$ counts the number of levels i whose cover sets C_i include new points that were absent on higher levels. Then $|H(\mathcal{T})| \leq |R|$ as any point can be alone at its own level.

Lemma B.8. *Any finite set R has the bound $|H(\mathcal{T}(R))| \leq 1 + \log_2(\Delta(R))$.* ■

Proof. We have $|H(\mathcal{T}(R))| \leq l_{\max} - l_{\min} + 1$ by Definition 2.6. We estimate $l_{\max} - l_{\min}$ as follows.

Let $p \in R$ be a point such that $\operatorname{diam}(R) = \max_{q \in R} d(p, q)$. Then R is covered by the closed ball $\bar{B}(p; \operatorname{diam}(R))$. Hence the cover set C_i at the level $i = \log_2(\operatorname{diam}(R))$ consists of a single point p. The separation condition in Definition 2.1 implies that $l_{\max} \leq \log_2(d_{\max}(R))$. Since any distinct points $p, q \in R$ have $d(p, q) \geq d_{\min}(R)$, the covering condition implies that no new points can enter the cover set C_i at the level $i = [\log_2(d_{\min}(R))]$, so $l_{\min} \geq \log_2(d_{\min}(R))$. Then $|H(\mathcal{T}(R))| \leq 1 + l_{\max} - l_{\min} \leq 1 + \log_2(\frac{\operatorname{diam}(R)}{d_{\min}(R)})$. □

If the aspect ratio $\Delta(R) = O(\operatorname{Poly}(|R|))$ polynomially depends on the size $|R|$, then $|H(\mathcal{T}(R))| \leq O(\log(|R|))$. Lemma 2.4 corresponds Beygelzimer et al. (2006a, Lemma 4.2) with slightly modified notation.

C. The minimized expansion constant in a normed vector space on $\mathbb{R}$

In this section, main Theorem C.15 will show that, for any finite subset R of a normed vector space $(\mathbb{R}^n, \|\cdot\|)$, the minimized expansion constant from Definition 1.4 has the upper bound 2^n, so

$$c_m(R) = \inf_{0<\xi} \inf_{R \subseteq A \subseteq \mathbb{R}^n} \sup_{p \in A, t > \xi} \frac{|\bar{B}(p,2t) \cap A|}{|\bar{B}(p,t) \cap A|} \leq 2^n.$$

The proof of Theorem C.15 is based on the volume argument. We briefly explain the idea before giving the proof later. For this purpose, we recall the definition of the Lebesgue measure in Definition C.2.

In Definition C.5 we define concepts of grid $G(\delta) = \delta \cdot \mathbb{Z}^n$ and cubic regions $\bar{V}(p,\delta) = p + [-\frac{\delta}{2}, \frac{\delta}{2}]^n$. For every $\delta > 0$ we define grid extension $U(\delta)$ of R as set $U(\delta) = (G(\delta) \setminus f(R)) \cup R$, where $f : R \to G(\delta)$ is used to replace points of R with their nearest neighbors in grid $G(\delta)$.

Note that ξ in the definition of $c_m(R)$ acts as a low bound for radius $t > \xi$. Let $\gamma > 0$ be a constant, that depends on dimension n and norm $\|\cdot\|$. In Lemma C.13 it is shown that if δ satisfies $0 < \delta < \frac{\xi}{\gamma}$, then for any $p \in U(\delta)$ and $t > \xi$ we can bound $|\bar{B}(p,t) \cap U(\delta)|$ as follows:

$$\frac{\mu(\bar{B}(p, t - \delta \cdot \gamma))}{\delta^n} \leq |\bar{B}(p,t) \cap U(\delta)| \leq \frac{\mu(\bar{B}(p, t + \delta \cdot \gamma))}{\delta^n}.$$

Therefore

$$\frac{|\bar{B}(p,2t) \cap U(\delta)|}{|\bar{B}(p,t) \cap U(\delta)|} \leq \frac{\mu(\bar{B}(p, 2t + \delta \cdot \gamma))}{\mu(\bar{B}(p, t - \delta \cdot \gamma))}.$$

Now since this inequality is satisfied for any $\delta > 0$, we can choose arbitrary dense grid extension $U(\delta)$. It will be shown that when $\delta \to 0$, then

$$\frac{\mu(\bar{B}(p, 2t + \delta \cdot \gamma))}{\mu(\bar{B}(p, t - \delta \cdot \gamma))} \to 2^n.$$

Then we can conclude that $c_m(R) \leq 2^n$.

Definition C.1 (Normed vector space $(\mathbb{R}^n, \|\cdot\|)$ on real numbers $\mathbb{R}$ Rudin (1990))**.** *Consider $\mathbb{R}^n$ as a vector space. A* norm *is a function $\|\cdot\| : \mathbb{R}^n \to \mathbb{R}$ satisfying the properties below.*

1. *Non-negativity : $\|x\| \geq 0$.*
2. *The norm is positive on nonzero vectors, so $\|x\| = 0$ implies that $x = 0$.*
3. *For every vector $x \in \mathbb{R}^n$, and every scalar $a \in \mathbb{R}$: $\|a \cdot x\| = |a| \cdot \|x\|$.*
4. *The triangle inequality holds for every $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^n$, $\|x + y\| \leq \|x\| + \|y\|$.*

A norm induces a metric by the formula $d(x,y) = \|x - y\|$. For every $i \in \{1,...,n\}$ let e_i be a unit vector of $\mathbb{R}^n$ i.e. $e_i(i) = 1$ and $e_i(j) = 0$ for all $j \in \{1,...,n\} \setminus \{i\}$. Define $\rho = \max_{i \in \{1,...,n\}} \|e_i\|$. ■

Definition C.2 (Lebesgue outer measure, Jones (2000, Section 2.A))**.** *Let $\mathbb{R}^n$ be an n-dimensional space. Define n-dimensional interval as*

$$I = \{x \in \mathbb{R}^n \mid a_i \leq x_i \leq b_i, i = 1,...,n\} = [a_1, b_1] \times ... \times [a_n, b_n],$$

with sides parallel to the coordinate axes. Define Lebesgue outer measure $\mu^* : \{A \mid A \subseteq \mathbb{R}^n\} \to [0,\infty) \cup \{\infty\}$ of interval I as $\mu^*(I) = (b_1 - a_1) \cdot ... \cdot (b_n - a_n)$. The Lebesgue μ measure of a set $A \subseteq \mathbb{R}^n$ is defined as:

$$\mu^*(A) = \inf_A \{\sum_{i=0}^{\infty} \mu^*(I_i) \mid A \subseteq \cup_{i=0}^{\infty} I_i\},$$

where the infinium is taken over all covering of A by countably many intervals $I_i, i = 1,2....$ If set $E \subseteq \mathbb{R}^n$ satisfies $\mu^*(A) = \mu^*(A \cap E) + \mu^*(A \setminus E)$ for all $A \subseteq \mathbb{R}^n$, then E is lebesgue-measurable and we set $\mu(E) = \mu^*(E)$. ■

It should be noted that all open sets and closed sets , as well as compact sets are Lebesgue-measurable.

Lemma C.3 (Basic properties of Lebesgue measure, Jones (2000, Section 2.A))**.** *A Lebesgue outer measure μ^* of Definition C.2 satisfies the following conditions:*

1. $\mu^*(\emptyset) = 0,$
2. $\mu^*(A) \leq \mu^*(B)$ *whenever* $A \subseteq B \subseteq \mathbb{R}^n$ *and*
3. $\mu^*(\cup_{i=1}^{\infty} \mu^*(A_i)) \leq \sum_{i=1}^{\infty} \mu^*(A_i).$

Lemma C.4 (Lebesgue measure scale property, Jones (2000, Section 3.B))**.** *Let μ be Lebesgue measure on a normed vector space $(\mathbb{R}^n, \|\cdot\|)$. Then, for any $p \in \mathbb{R}^n$ and $t \in \mathbb{R}_+$, we have: $\mu(\bar{B}(p,t)) = t^n \cdot \mu(\bar{B}(p,1))$.*

Definition C.5 (Grid and Cubic region)**.** *Let $\mathbb{R}^n$ be a normed vector space and let $\delta \in \mathbb{R}$. Define δ-grid on $\mathbb{R}^n$ as the set $G(\delta) = \{t \cdot \delta \mid t \in \mathbb{Z}^n\}$. For any $p \in \mathbb{R}^n$ define its open cubic region $V(p,\delta) \subseteq \mathbb{R}^n$ as the set $\{p + u \mid u \in (-\frac{\delta}{2}, \frac{\delta}{2})^n\}$ and closed cubic region $\bar{V}(p,\delta) \subseteq \mathbb{R}^n$ as $\{p + u \mid u \in [-\frac{\delta}{2}, \frac{\delta}{2}]^n\}$.*

Note that the union $\cup_{p \in G(\delta)} V(p,\delta)$ covers whole set $\mathbb{R}^n$.

Lemma C.6 (Cubic regions are separate)**.** *In conditions of Definition C.5 let $p, q \in G(\delta)$ be distinct points. Then their cubic regions are separate i.e. $V(p,\delta) \cap V(q,\delta) = \emptyset$.*

Proof. Assume contrary that there exists $a \in V(p,\delta) \cap V(q,\delta)$, then $|a(i) - p(i)| < \frac{\delta}{2}$ and $|a(i) - q(i)| < \frac{\delta}{2}$ for all $i \in \{1,...,n\}$. Since $p \neq q$, there exists index j, such that $p(j) \neq q(j)$. By definition of grid $G(\delta)$ it follows that $|p(j) - q(j)| \geq \delta$. However, by triangle inequality we have

$$|p(j) - q(j)| \leq |p(j) - a(j)| + |q(j) - a(j)| < \frac{\delta}{2} + \frac{\delta}{2} = \delta,$$

which is a contradiction. Therefore $V(p,\delta) \cap V(q,\delta) = \emptyset$.

□

Lemma C.7. *Let $\mathbb{R}^n$ be a normed vector space of Definition C.1. Let $\delta \in \mathbb{R}$ and let $G(\delta)$ be a grid of Definition C.5. Let $p \in G(\delta)$ and let $q \in V(p,\delta)$, then $d(p,q) \leq \frac{n \cdot \delta \cdot \rho}{2}$*

Proof. Let $\gamma \in \mathbb{R}$ be such that $q = p + \gamma$. By condition (3) of Definition C.1 we have $\|\gamma(i)\| \leq \|e_i\| \cdot \frac{\delta}{2} \leq \frac{\delta \cdot \rho}{2}$ for all $i \in \{1,...,n\}$. By the definition of norm and triangle inequality we have:

$$d(p,q) = \|p - q\| = \|\gamma\| \leq \sum_{i=1}^{n} \|\gamma(i)\| \leq \frac{n \cdot \delta \cdot \rho}{2}.$$

□

Any normed vector space $(\mathbb{R}^n, \|\cdot\|)$ has the metric $d(x,y) = \|x - y\|$.

Lemma C.8 (Existence of covering grid)**.** *Let R be a finite subset of a normed vector space $(\mathbb{R}^n, \|\cdot\|)$. Then for any $\delta \in \mathbb{R}$ having $\delta < \frac{d_{\min}(R)}{n \cdot \rho}$, then any map $f : R \to G(\delta)$ which maps $p \in R$ to one of its nearest neighbor in $G(\delta)$ is a well-defined injection.*

Proof. Let f be an arbitrary map $f : R \to G(\delta)$ mapping point $p \in R$ to one of its nearest neighbors. This map is clearly well-defined. Let us now show that it is injective. Assume that $x, y \in R$ are such that $f(x) = f(y)$. Then by triangle inequality and Lemma C.7 we have:

$$d(x,y) \leq d(x,p) + d(p,y) \leq n \cdot \delta \cdot \rho < d_{\min}(R),$$

it follows that $x = y$. Therefore map f is injective.

□

Lemma C.9. *Let R be a finite subset of normed space $(\mathbb{R}^n, d)$, let ρ be as in Definition C.1 and let $\delta \in \mathbb{R}$ be such that $0 < \delta < \frac{d_{\min}(R)}{n \cdot \rho}$. Let $p \in R$ be arbitrary point and let $t > \frac{n \cdot \delta \cdot \rho}{2}$ be a real number. Then there exists a set $U(\delta)$ satisfying $R \subseteq U(\delta)$ and*

$$|G(\delta) \cap \bar{B}(p, t - \frac{n \cdot \delta \cdot \rho}{2})| \leq |U(\delta) \cap \bar{B}(p, t)| \leq |G(\delta) \cap \bar{B}(p, t + \frac{n \cdot \delta \cdot \rho}{2})|$$

Proof. Let $f : R \to G(\delta)$ be an injection from Lemma C.8, which maps every $q \in R$ to one of its nearest neighbors in $G(\delta)$. Define $U(\delta) = (G(\delta) \setminus f(R)) \cup R$. Let us first show that

$$g : U(\delta) \cap \bar{B}(p, t) \to G(\delta) \cap \bar{B}(p, t + \frac{n \cdot \delta \cdot \rho}{2}),$$

defined by $g(q) = f(q)$, if $q \in R$ and $g(q) = q$, if $q \notin R$, is an injection. Let us show first that the map g is well-defined, if $q \notin R$, the claim is trivial. Let $q \notin R$, then by triangle inequality $d(g(q), p) \leq d(q, p) + d(g(q), q) \leq t + \frac{n \cdot \delta \cdot \rho}{2}$. Assume now that $g(a) = g(b)$ for some $a, b \in U(\delta) \cap \bar{B}(p, t)$. Let us first show that either a, b both belong to R or neither of a, b belong to R. Assume contrary that $a \in R$ and $b \notin R$. Since $b \notin R$ we have $b \in G(\delta) \setminus f(R)$. On the other hand since $h(a) = h(b)$ we have $f(a) = b$, therefore $b \in f(R)$, which is a contradiction. If both, a and b belong to R we have $a = b$, similarly if $a, b \notin R$ we have $a = b$ by injectivity of function f. Therefore we have now shown that g is well-defined injection. It follows $|U(\delta) \cap \bar{B}(p, t)| \leq |G(\delta) \cap \bar{B}(p, t + \frac{n \cdot \delta \cdot \rho}{2})|$. Let us now show that map

$$h : G(\delta) \cap \bar{B}(p, t - \frac{n \cdot \delta \cdot \rho}{2}) \to U(\delta) \cap \bar{B}(p, t),$$

defined by $h(q) = f^{-1}(q)$, if $q \in f(R)$ and $h(q) = q$, if $q \notin f(R)$ is well-defined injection. Let us first show that the map is well-defined. Let $q \in G(\delta) \cap \bar{B}(p, t - \frac{n \cdot \delta \cdot \rho}{2})$, if $q \notin f(R)$ the claim is satisfied trivially. If $q \in f(R)$, then by definition $d(h(q), q) \leq \frac{n \cdot \delta \cdot \rho}{2}$. By using triangle inequality we obtain:

$$d(p, h(q)) \leq d(p, q) + d(q, h(q)) \leq t - \frac{n \cdot \delta \cdot \rho}{2} + \frac{n \cdot \delta \cdot \rho}{2} \leq t.$$

Therefore $h(q) \in U(\delta) \cap \bar{B}(p, t)$.

Let us now show that h is an injection. Let $a, b \in G(\delta) \cap \bar{B}(p, t - \frac{n \cdot \delta}{2})$ assume that $h(a) = h(b)$, let us show that $a = b$. Let us first show that either $a, b \in f(R)$ or neither of a, b belong to $f(R)$. Assume contrary that $a \in f(R)$ and $b \notin f(R)$. Then $h(a) = h(b)$ implies that $f^{-1}(a) = b$. Since $f^{-1}(a) \in R$, we have $b \in R$. Since $b \in G(\delta)$, it follows that $f(b) = b$, which is a contradiction since $b \notin f(R)$. Assume now that $a, b \in f(R)$, then the claim follows by noting that f^{-1} is injection. If $a, b \notin f(R)$, then claim follows by noting that $h(a) = a$ and $h(b) = b$. Therefore map h is injection. It follows that $|G(\delta) \cap \bar{B}(p, t - \frac{n \cdot \delta \cdot \rho}{2})| \leq |U(\delta) \cap \bar{B}(p, t)|$.

$\square$

Lemma C.10. *Let R be a finite subset of normed vector space $\mathbb{R}^n$ and let $\delta \in \mathbb{R}$. For any $p \in G(\delta)$ recall that $V(p, \delta)$ is Minkowski sum $p + (-\frac{\delta}{2}, \frac{\delta}{2})^n$. Define*

$$\bar{W}(p, t, \delta) = \bigcup_{q \in \bar{B}(p,t) \cap G(\delta)} \bar{V}(q, \delta).$$

Then for any $p \in R$ and $t > \frac{n \cdot \delta \cdot \rho}{2}$ we have:

$$\bar{B}(p, t - \frac{n \cdot \delta \cdot \rho}{2}) \subseteq \bar{W}(p, t, \delta) \subseteq \bar{B}(p, t + \frac{n \cdot \delta \cdot \rho}{2}).$$

Proof. Let $u \in \bar{B}(p, t - \frac{n \cdot \delta \cdot \rho}{2})$ be an arbitrary point. Since $\{\bar{V}(q, \delta) \mid q \in G(\delta)\}$ covers R it follows that there exists $a \in G(\delta)$ such that $u \in \bar{V}(a, \delta)$. By triangle inequality we obtain:

$$d(a, p) \leq d(a, u) + d(u, p) \leq \frac{n \cdot \delta \cdot \rho}{n} + t - \frac{n \cdot \delta \cdot \rho}{n} \leq t.$$

It follows that $\bar{V}(w,\delta) \in \bar{W}(p,t)$, therefore $p \in \bar{W}(p,t,\delta)$. We have $\bar{B}(p,t-\frac{n\cdot\delta\cdot\rho}{2}) \subseteq \bar{W}(p,t,\delta)$. Let $u \in \bar{W}(p,t,\delta)$, then there exists $a \in G(\delta)$ such that $u \in \bar{V}(a,\delta)$ and $\bar{V}(a,\delta) \in \bar{W}(p,t)$. By triangle inequality we obtain:

$$d(u,p) \leq d(u,a) + d(a,p) \leq \frac{n\cdot\delta\cdot\rho}{n} + t.$$

It follows that $u \in \bar{B}(p,t+\frac{n\cdot\delta\cdot\rho}{2})$. Therefore $\bar{W}(p,t,\delta) \subseteq \bar{B}(p,t+\frac{n\cdot\delta\cdot\rho}{2})$ which proves the claim.

□

Lemma C.11 (Countable additivity, Jones (2000, Section 2.A))**.** *Assume that $A_i \subseteq \mathbb{R}^n$, $i = 1,2,...$, are pairwise disjoint i.e. $A_i \cap A_j = \emptyset$ for all $i \neq j$ Lebesgue-measurable sets. Then*

$$\mu(\bigcup_{i=0}^{\infty} A_i) = \sum_{i=1}^{\infty} \mu(A_i).$$

Lemma C.12 (Lebesgue measure of $\bar{W}(p,t,\delta)$)**.** *In notations of Lemma C.10 let μ be a Lebesgue measure on R from Definition C.2, then $\mu(\bar{W}(p,t,\delta)) = \delta^n \cdot |\bar{B}(p,t) \cap G(\delta)|$.*

Proof. Define $W(p,t,\delta) = \bigcup_{q \in \bar{B}(p,t)\cap G(\delta)} V(q,\delta)$. Recall that for all $p \in \mathbb{R}^n$ and $\delta > 0$ set $\bar{V}(p,t)$ is a closed $n-$dimensional interval and $V(p,t)$ is an open n-dimensional interval. Therefore we have $\mu(\bar{V}(p,t)) = \mu(V(p,t))$. Since $\bar{V}(p,t)$ is a closed interval, it follows that $\mu(\bar{V}(p,t)) = \delta^n$. Since all the sets of W are separate we can use Lemma C.11 to obtain:

$$\mu(W(p,t,\delta)) = \sum_{A \in W(p,t)} \mu(A) = \sum_{A \in \bar{W}(p,t)} \mu(A) = \delta^n \cdot |\bar{B}(p,t) \cap G(\delta)|$$

By Lemma C.3 (2), since $W(p,t,\delta) \subseteq \bar{W}(p,t,\delta)$ we obtain $\mu(\cup\bar{W}(p,t)) \geq \delta^n \cdot |\bar{B}(p,t) \cap G(\delta)|$. On the other hand, by Lemma C.3 (3) we obtain

$$\mu(\bar{W}(p,t,\delta)) \leq \sum_{A \in \bar{W}(p,t)} \mu(A) = \delta^n \cdot |\bar{B}(p,t) \cap G(\delta)|$$

Therefore we have shown that $\mu(\bar{W}(p,t,\delta)) = \delta^n \cdot |\bar{B}(p,t) \cap G(\delta)|$. □

Lemma C.13 (Set $U(\delta)$ bounds)**.** *Let $\mathbb{R}^n$ be a normed vector space. Let $R \subseteq \mathbb{R}^n$ be its finite subset. Then any set $U(\delta)$ of Lemma C.9 satisfies the following inequalities:*

$$\frac{\mu(\bar{B}(p,t-\delta\cdot n\cdot\rho))}{\delta^n} \leq |\bar{B}(p,t) \cap U(\delta)| \leq \frac{\mu(\bar{B}(p,t+\delta\cdot\gamma\cdot n\cdot\rho))}{\delta^n},$$

for all $p \in R$ and $t > n\cdot\delta\cdot\rho$.

Proof. Let $p \in \mathbb{R}^n$ be an arbitrary point and let $t > n\cdot\delta\cdot\rho$ be an arbitrary real number. By Lemma C.9 it follows:

$$|G(\delta) \cap \bar{B}(p,t+\frac{n\cdot\delta\cdot\rho}{2})| \leq |\bar{B}(p,t) \cap U(\delta)| \leq |G(\delta) \cap \bar{B}(p,t+\frac{n\cdot\delta\cdot\rho}{2})|.$$

Let $\bar{W}(p,t+\frac{n\cdot\delta\cdot\rho}{2},\delta) = \cup_q\{\bar{V}(q,\delta) \mid q \in \bar{B}(p,t+\frac{n\cdot\delta\cdot\rho}{2})\}$. By Lemma C.10 we have:

$$\bar{B}(p,t-n\cdot\delta\cdot\rho) \subseteq \bar{W}(p,t-\frac{n\cdot\delta\cdot\rho}{2},\delta) \text{ and } \bar{W}(p,t+\frac{n\cdot\delta\cdot\rho}{2},\delta) \subseteq \bar{B}(p,t+n\cdot\delta\cdot\rho)$$

By Lemma C.3 we have $\mu(\bar{W}(p,t+\frac{n\cdot\delta\cdot\rho}{2},\delta)) \leq \mu(\bar{B}(p,t+n\cdot\delta\cdot\rho))$. By Lemma C.12 we have:

$$\mu(\bar{W}(p,t+\frac{n\cdot\delta\cdot\rho}{2},\delta)) = \delta^n \cdot |\bar{B}(p,t+\frac{n\cdot\delta\cdot\rho}{2}) \cap G(\delta)|.$$

By combining the facts we obtain:

$$|\bar{B}(p,t) \cap U(\delta)| \leq |G(\delta) \cap \bar{B}(p,t+\frac{n\cdot\delta\cdot\rho}{2})| \leq \frac{\mu(\bar{W}(p,t+\frac{n\cdot\delta\cdot\rho}{2},\delta))}{\delta^n} \leq \frac{\mu(\bar{B}(p,t+n\cdot\delta\cdot\rho))}{\delta^n}$$

$$|\bar{B}(p,t)\cap U(\delta)| \geq |G(\delta)\cap \bar{B}(p,t-\frac{n\cdot\delta\cdot\rho}{2})| \geq \frac{\mu(\bar{W}(p,t-\frac{n\cdot\delta\cdot\rho}{2},\delta))}{\delta^n} \geq \frac{\mu(\bar{B}(p,t-n\cdot\delta\cdot\rho))}{\delta^n}$$

which concludes the proofs.

□

Lemma C.14 (Set $U(\delta)$ is locally finite)**.** *Let $\mathbb{R}^n$ be a normed vector space. Let $R\subseteq\mathbb{R}^n$ be its finite subset Then any set $U(\delta)$ from Lemma C.9 is locally finite.*

Proof. With the exact same proof of Lemma C.13 it can be shown that

$$|\bar{B}(p,t)\cap U(\delta)| \leq \frac{\mu(\bar{B}(p,t+\delta\cdot n\cdot\rho))}{\delta^n}$$

is satisfied for all $p\in R$ and $t>0$. Therefore $|\bar{B}(p,t)\cap U(\delta)|$ is finite as well.

□

Recall that *minimized expansion constant* of Definition 1.4 of a finite subset R of a metric space (X,d) was defined as $c_m(R)=\lim\limits_{\xi\to 0^+}\inf\limits_{R\subseteq A\subseteq X}\sup\limits_{p\in A,t>\xi}\dfrac{|\bar{B}(p,2t)\cap A|}{|\bar{B}(p,t)\cap A|}$ where A is a locally finite set which covers R.

Theorem C.15 (The minimized expansion constant of a finite subset R of $\mathbb{R}^n$ is at most 2^n)**.** *Let R be a finite subset of a normed Euclidean space $\mathbb{R}^n$. Let $c_m(R)$ be the minimized expansion constant of Definition 1.4, then $c_m(R)\leq 2^n$.*

Proof. Let $0<\xi<\frac{d_{\min}(R)}{2}$ be an arbitrary real number. Let $0<\delta<\frac{\xi}{n\cdot\rho}$ be a real number. Since $\delta<\frac{d_{\min}(R)}{2\cdot n\cdot\rho}$ by Lemma C.13 we have:

$$\frac{\mu(\bar{B}(p,t-\delta\cdot n\cdot\rho))}{\delta^n} \leq |\bar{B}(p,t)\cap U(\delta)| \leq \frac{\mu(\bar{B}(p,t+\delta\cdot\gamma\cdot n\cdot\rho))}{\delta^n}$$

Note that by Lemma C.4 we have: $\mu(\bar{B}(q,y))=y^n\cdot\mu(\bar{B}(q,1))$ for any $q\in\mathbb{R}^n$ and $y\in\mathbb{R}_+$. Therefore

$$\frac{|\bar{B}(p,2t)\cap U(\delta)|}{|\bar{B}(p,t)\cap U(\delta)|} \leq \frac{\mu(\bar{B}(p,2t+n\delta\rho))\cdot\delta^2}{\mu(\bar{B}(p,t-n\delta\rho))\cdot\delta^2} = \frac{(2t+n\delta\rho)^n\cdot\mu(\bar{B}(p,1))}{(t-n\delta\rho)^n\cdot\mu(\bar{B}(p,1))} = \frac{(2t+n\delta\rho)^n}{(t-n\delta\rho)^n},$$

is satisfied for for all $t>\xi$. Since $0<\xi<\frac{d_{\min}(R)}{2}$ was chosen arbitrarily, we conclude that:

$$c_m(R)=\lim_{\xi\to 0^+}\inf_{R\subseteq A\subseteq X}\sup_{p\in A,t>\xi}\frac{|\bar{B}(p,2t)|\cap A}{|\bar{B}(p,t)|\cap A} \leq \lim_{\delta\to 0}\frac{|\bar{B}(p,2t)\cap U(\delta)|}{|\bar{B}(p,t)\cap U(\delta)|} = \lim_{\delta\to 0}\frac{(2t+\delta\cdot n\rho)^n}{(t-\delta\cdot n\rho)^n} = \frac{2^n\cdot t^n}{t^n} = 2^n.$$

□

D. Distinctive descendant sets

This section introduces auxiliary concepts for future proofs. The main concept is a distinctive descendant set in Definition 2.8. The distinctive descendant set at a level i of a node $p\in\mathcal{T}(R)$ in a compressed cover tree corresponds to the set of descendants of a copy of node p at level i in the original implicit cover tree $T(R)$. Other important concepts are λ-point of Definition D.6 that is used in Algorithm F.2 as an approximation for k-nearest neighboring point. The β-point property of λ-point defined in Lemma D.15 plays a major role in the proof of the main worst-case time complexity result Theorem 4.9.

Definition 2.8 (Distinctive descendant sets)**.** *Let $R\subseteq X$ be a finite reference set with a compressed cover tree $\mathcal{T}(R)$. For any node $p\in\mathcal{T}(R)$ and level $i\leq l(p)-1$, set $V_i(p)=\{u\in\text{Descendants}(p)\mid i\leq l(u)\leq l(p)-1\}$. If $i\geq l(p)$, then set $V_i(p)=\emptyset$. For any level $i\leq l(p)$, the* distinctive descendant set *is $\mathcal{S}_i(p,\mathcal{T}(R))=\text{Descendants}(p)\setminus\bigcup\limits_{u\in V_i(p)}\text{Descendants}(u)$ and has the size $|\mathcal{S}_i(p,\mathcal{T}(R))|$.*

Lemma D.1 (Distinctive descendant set inclusion property)**.** *In conditions of Definition 2.8 let $p\in R$ and let i,j be integers satisfying $l_{\min}(\mathcal{T}(R))\leq i\leq j\leq l(p)-1$. Then $\mathcal{S}_i(p,\mathcal{T}(R))\subseteq\mathcal{S}_j(p,\mathcal{T}(R))$.*

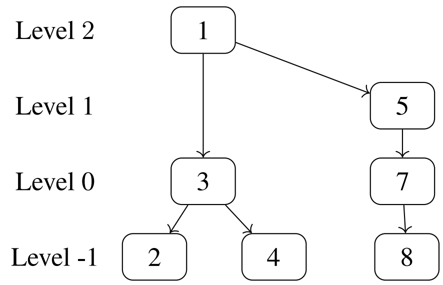

Figure 9. Consider a compressed cover tree $\mathcal{T}(R)$ that was built on set $R = \{1, 2, 3, 4, 5, 7, 8\}$. Let $\mathcal{S}_i(p, \mathcal{T}(R))$ be a distinctive descendant set of Definition 2.8. Then $V_2(1) = \emptyset, V_1(1) = \{5\}$ and $V_0(1) = \{3, 5, 7\}$. And also $\mathcal{S}_2(1, \mathcal{T}(R)) = \{1, 2, 3, 4, 5, 7, 8\}$, $\mathcal{S}_1(1, \mathcal{T}(R)) = \{1, 2, 3, 4\}$ and $\mathcal{S}_0(1, \mathcal{T}(R)) = \{1\}$.

Essential levels of a node $p \in \mathcal{T}(R)$ have 1-1 correspondence to the set consisting of all nodes containing p in the explicit representation of cover tree in (Beygelzimer et al., 2006a), see Figure 6 middle.

Definition D.2 (Essential levels of a node)**.** *Let $R \subseteq X$ be a finite reference set with a cover tree $\mathcal{T}(R)$. Let $q \in \mathcal{T}(R)$ be a node. Let (t_i) for $i \in \{0, 1, ..., n\}$ be a sequence of $H(\mathcal{T}(R))$ in such a way that $t_0 = l(q)$, $t_n = l_{\min}(\mathcal{T}(R))$ and for all i we have $t_{i+1} = \mathrm{Next}(q, t_i, \mathcal{T}(R))$. Define the set of essential indices $\mathcal{E}(q, \mathcal{T}(R)) = \{t_i \mid i \in \{0, ..., n\}\}$.* ■

Lemma D.3 (Number of essential levels)**.** *Let $R \subseteq X$ be a finite reference set with a cover tree $\mathcal{T}(R)$. Then $\sum_{p \in R} |\mathcal{E}(p, \mathcal{T}(R))| \leq 2 \cdot |R|$, where $\mathcal{E}(p, \mathcal{T}(R))$ appears in Definition D.2.* ■

Proof. Let us prove this claim by induction on size $|R|$. In basecase $R = \{r\}$ and therefore $|\mathcal{E}(r, \mathcal{T}(R))| = 1$. Assume now that the claim holds for any tree $\mathcal{T}(R)$, where $|R| = m$ and let us prove that if we add any node $v \in X \setminus R$ to tree $\mathcal{T}(R)$, then $\sum_{p \in R} |\mathcal{E}(p, \mathcal{T}(R \cup \{v\}))| \leq 2 \cdot |R| + 2$. Assume that we have added u to $\mathcal{T}(R)$, in such a way that v is its new parent. Then $|\mathcal{E}(p, \mathcal{T}(R \cup \{v\}))| = |\mathcal{E}(p, \mathcal{T}(R))| + 1$ and $|\mathcal{E}(v, \mathcal{T}(R \cup \{v\}))| = 1$. We have:

$$\sum_{p \in R \cup \{u\}} |\mathcal{E}(p, \mathcal{T}(R))| = \sum_{p \in R} |\mathcal{E}(p, \mathcal{T}(R))| + 1 + |\mathcal{E}(v, \mathcal{T}(R \cup \{v\}))| \leq 2 \cdot |R| + 2 \leq 2(|R \cup \{v\}|)$$

which completes the induction step. □

Algorithm D.4 This algorithm returns sizes of distinctive descendant set $\mathcal{S}_i(p, \mathcal{T}(R))$ for all essential levels $i \in \mathcal{E}(p, \mathcal{T}(R))$

Function : CountDistinctiveDescendants(Node p, a level i of $\mathcal{T}(R)$)
Output : an integer
if $i > l_{\min}(\mathcal{T}(Q))$ **then**
  for $q \in$ Children(p) having $l(p) = i - 1$ or $q = p$ **do**
    Set $s = 0$
    $j \leftarrow 1 + \mathrm{Next}(q, i - 1, \mathcal{T}(R))$
    $s \leftarrow s+$CountDistinctiveDescendants(q, j)
  end for
else
  Set $s = 1$
end if
Set $|\mathcal{S}_i(p)| = s$ and **return** s

Lemma D.5. *Let R be a finite subset of a metric space. Let $\mathcal{T}(R)$ be a compressed cover tree on R. Then, Algorithm D.4 computes the sizes $|\mathcal{S}_i(p, \mathcal{T}(R))|$ for all $p \in R$ and essential levels $i \in \mathcal{E}(p, \mathcal{T}(R))$ in time $O(|R|)$.* ■

Proof. By Lemma D.3 we have $\sum_{p \in R} |\mathcal{E}(p, \mathcal{T}(R))| \leq 2 \cdot |R|$. Since CountDistinctiveDescendants is called once for every any combination $p \in R$ and $i \in \mathcal{E}(p, \mathcal{T}(R))$ it follows that the time complexity of Algorithm D.4 is $O(R)$.

□

Recall that the neighbor set $N(q;r) = \{p \in C \mid d(q,p) \le d(q,r)\}$ was introduced in Definition 1.2.

Definition D.6 (λ-point)**.** *Fix a query point q in a metric space (X,d) and fix any level $i \in \mathbb{Z}$. Let $\mathcal{T}(R)$ be its compressed cover tree on a finite reference set $R \subseteq X$. Let C be a subset of a cover set C_i from Definition 2.1 satisfying $\sum_{p\in C} |\mathcal{S}_i(p,\mathcal{T}(R))| \ge k$, where $\mathcal{S}_i(p,\mathcal{T}(R))$ is the distinctive descendant set from Definition 2.8. For any $k \ge 1$, define $\lambda_k(q,C)$ as a point $\lambda \in C$ that minimizes $d(q,\lambda)$ subject to $\sum_{p\in N(q;\lambda)} |\mathcal{S}_i(p,\mathcal{T}(R))| \ge k$.* ■

Algorithm D.7 Finding k-lowest element of a finite subset $A \subseteq R$ with priority function $f : A \to \mathbb{R}$

Input: Ordered subset $A \subseteq R$, priority function $f : A \to \mathbb{R}$, an integer $k \in \mathbb{Z}$
Initialize an empty max-binary heap B and an empty array D on points A.
for $p \in A$ **do**
    add p to B with priority $f(p)$
    if $|H| \ge k$ **then**
        remove the point with a maximal value from B
    end if
end for
Transfer points from the binary heap B to the array D in reverse order.
return D.

Algorithm D.8 Computation of a λ-point of Definition D.6 in line 6 of Algorithm F.2

1: **Input:** A point $q \in X$, a subset C of a level set C_i of a compressed cover tree $\mathcal{T}(R)$, an integer $k \in \mathbb{Z}$
2: Define $f : C \to \mathbb{R}$ by setting $f(p) = d(p,q)$.
3: Run Algorithm D.7 on inputs (C,f,k) and retrieve array D.
4: Find the smallest index j such that $\sum_{t=0}^{j} |\mathcal{S}_i(D[t],\mathcal{T}(R))| \ge k$.
5: **return** $\lambda = D[j]$.

Lemma D.9. *Let $A \subseteq R$ be a finite subset and let $f : A \to \mathbb{R}$ be a priority function and let $k \in \mathbb{Z}_+$. Then Algorithm D.7 finds k-smallest elements of A in time $|A| \cdot \log_2(k)$*

Proof. Adding and removing element from binary heap data structure Cormen (1990, section 6.5) takes at most $O(\log(n))$ time, where n is the size of binary heap. Since the size of our binary heap is capped at k and we add/remove at most $|A|$ elements, the total time complexity is $O(|A| \cdot \log_2(k))$. □

Lemma D.10 (time complexity of a λ-point)**.** *In the conditions of Definition D.6, the time complexity of Algorithm D.8 is $O(|C| \cdot \log_2(k))$.*

Proof. Note that in line 4 we have $|\mathcal{S}_i(D[t],\mathcal{T}(R))| \ge 1$ for all $t = 0,...,j$. Therefore the time complexity of line 4 is $O(k)$. By Lemma D.9 The time complexity of line 3 is $O(|C| \cdot \log_2(k))$, which proves the claim. □

Lemma D.11 (separation)**.** *In the conditions of Definition 2.8, let $p \ne q$ be nodes of $\mathcal{T}(R)$ with $l(p) \ge i$, $l(q) \ge i$. Then $\mathcal{S}_i(p,\mathcal{T}(R)) \cap \mathcal{S}_i(q,\mathcal{T}(R)) = \emptyset$.* ■

Proof. Without loss of generality assume that $l(p) \ge l(q)$. If q is not a descendant of p, the lemma holds trivially due to $\text{Descendants}(q) \cap \text{Descendants}(p) = \emptyset$. If q is a descendant of p, then $l(q) \le l(p) - 1$ and therefore $q \in V_i(p)$. It follows that $\mathcal{S}_i(p,\mathcal{T}(R)) \cap \text{Descendants}(q) = \emptyset$ and therefore

$$\mathcal{S}_i(p,\mathcal{T}(R)) \cap \mathcal{S}_i(q,\mathcal{T}(R)) \subseteq \mathcal{S}_i(p,\mathcal{T}(R)) \cap \text{Descendants}(q) = \emptyset.$$

□

Lemma D.12 (Sum lemma)**.** *In the notations of Definition 2.8 assume that i is arbitrarily index and a subset $V \subseteq R$ satisfies $l(p) \ge i$ for all $p \in V$. Then*

$$|\bigcup_{p\in V} \mathcal{S}_i(p,\mathcal{T}(R))| = \sum_{p\in V} |\mathcal{S}_i(p,\mathcal{T}(R))|.$$

Proof. Proof follows from Lemma D.11. $\square$

By Lemma D.12 in Definition D.6 one can assume that $|\bigcup_{p \in C} \mathcal{S}_i(p, \mathcal{T}(R))| \geq k$.

Lemma 2.9. *Let $R \subseteq X$ be a finite reference set with a cover tree $\mathcal{T}(R)$. In the notations of Definition 2.8, let $p \in \mathcal{T}(R)$ be any node. If $w \in \mathcal{S}_i(p, \mathcal{T}(R))$ then either $w = p$ or there exists $a \in \text{Children}(p) \setminus \{p\}$ such that $l(a) < i$ and $w \in \text{Descendants}(a)$.*

Proof. Let $w \in \mathcal{S}_i(p)$ be an arbitrary node satisfying $w \neq p$. Let s be the node-to-root path of w. The inclusion $\mathcal{S}_i(p) \subseteq \text{Descendants}(p)$ implies that $w \in \text{Descendants}(p)$. Let $a \in \text{Children}(p) \setminus \{p\}$ be a child on the path s. If $l(a) \geq i$ then $a \in V_i(p)$. Note that $w \in \text{Descendants}(a)$. Therefore $w \notin \mathcal{S}_i(p)$, which is a contradiction. Hence $l(a) < i$. $\square$

Lemma D.13. *In the notations of Definition 2.8, let $p \in \mathcal{T}(R)$ be any node. If $w \in \mathcal{S}_i(p, \mathcal{T}(R))$ then $d(w, p) \leq 2^{i+1}$.* $\blacksquare$

Proof. By Lemma 2.9 either $w = \gamma$ or $w \in \text{Descendants}(a)$ for some $a \in \text{Children}(\gamma) \setminus \{\gamma\}$ for which $l(a) < i$. If $w = \gamma$, then trivially $d(\gamma, w) \leq 2^i$. Else w is a descendant of a, which is a child of node γ on level $i - 1$ or below, therefore by Lemma B.6 we have $d(\gamma, w) \leq 2^i$ anyway. $\square$

Lemma D.14. *Let R be a finite subset of a metric pace. Let $\mathcal{T}(R)$ be a compressed cover tree on R. Let $R_j \subseteq C_j$, where C_j is the ith cover set of $\mathcal{T}(R)$. Let $i = \max_{p \in R_j} \text{Next}(p, j, \mathcal{T}(R))$. Set $\mathcal{C}_i(R_j) = R_j \cup \{a \in \text{Children}(p)$ for some $p \in R_i \mid l(a) = i\}$. Then*

$$\bigcup_{p \in \mathcal{C}_i(R_j)} \mathcal{S}_i(p, \mathcal{T}(R)) = \bigcup_{p \in R_j} \mathcal{S}_j(p, \mathcal{T}(R)).$$

Proof. Let $a \in \bigcup_{p \in \mathcal{C}_i(R_j)} \mathcal{S}_i(p, \mathcal{T}(R))$ be an arbitrary node. Then there exits $u \in \mathcal{C}_i(R_j)$ having $a \in \mathcal{S}_i(u, \mathcal{T}(R))$. By definition of index i, either $u \in R_j$ or u has a parent in R_j. If $u \in R_j$ then we note that $V_j(u) \subseteq V_i(u)$. Since $a \notin V_i(u)$, we also have $a \notin V_j(u)$.

Otherwise let w be a parent of u. Therefore there are no descendants of w in having level in interval $[l(u) + 1, l(p) - 1]$. Since $l(u) = i$ and $j > i$ it follows that $V_j(w) = \emptyset$. Denote w to be the lowest level ancestor of u on level j. By cases above we have $a \notin V_j(w)$. Therefore it follows that

$$a \in \mathcal{S}_j(w, \mathcal{T}(R)) \subseteq \bigcup_{p \in R_j} \mathcal{S}_j(p, \mathcal{T}(R)).$$

To prove the converse inclusion assume now that $a \in \bigcup\limits_{p \in R_j} \mathcal{S}_j(p, \mathcal{T}(R))$. Then $a \in \mathcal{S}_j(v, \mathcal{T}(R))$ for some $w \in R_j$. Assume that w has no children at the level i. Then $V_j(w) = V_i(w)$ and

$$a \in \mathcal{S}_i(w, \mathcal{T}(R)) \subseteq \bigcup_{p \in \mathcal{C}_i(R_j)} \mathcal{S}_i(p, \mathcal{T}(R)).$$

Assume now that w has children at the level i. If there exists $b \in \text{Children}(w)$ for which $a \in \text{Descendants}(b)$. Since $V_i(b) = \emptyset$, we conclude that

$$a \in \mathcal{S}_i(b, \mathcal{T}(R)) \subseteq \bigcup_{p \in \mathcal{C}_i(R_j)} \mathcal{S}_i(p, \mathcal{T}(R)).$$

Assume that $a \notin \text{Descendants}(b)$ for all $b \in \text{Children}(w)$ with $l(b) = i$. Then $a \in \text{Descendants}(w)$ and $a \notin \text{Descendants}(b')$ for any $b' \in V_j(w)$. Then $a \in \mathcal{S}_i(w, \mathcal{T}(R))$ and the proof finishes:

$$\bigcup_{p \in R_j} \mathcal{S}_j(p, \mathcal{T}(R)) \subseteq \bigcup_{p \in \mathcal{C}_i(R_j)} \mathcal{S}_i(p, \mathcal{T}(R)).$$

$\square$

Lemma D.15 (β-point)**.** *In the notations of Definition D.6, let $C \subseteq C_i$ so that $\cup_{p \in C}\mathcal{S}_i(p, \mathcal{T}(R))$ contains all k-nearest neighbors of q. Set $\lambda = \lambda_k(q, C)$. Then R has a point β among the first k nearest neighbors of q such that $d(q, \lambda) \leq d(q, \beta) + 2^{i+1}$.* ■

Proof. We show that R has a point β among the first k nearest neighbors of q such that

$$\beta \in \bigcup_{p \in C} \mathcal{S}_i(p, \mathcal{T}(R)) \setminus \bigcup_{p \in N(q,\lambda) \setminus \{\lambda\}} \mathcal{S}_i(p, \mathcal{T}(R)).$$

Lemma D.12 and Definition D.6 imply that

$$|\bigcup_{p \in N(q,\lambda) \setminus \{\lambda\}} \mathcal{S}_i(p, \mathcal{T}(R))| = \sum_{p \in N(q,\lambda) \setminus \{\lambda\}} |\mathcal{S}_i(p, \mathcal{T}(R))| < k.$$

Since $\cup_{p \in C}\mathcal{S}_i(p, \mathcal{T}(R))$ contains all k-nearest neighbors of q, a required point $\beta \in R$ exists.

Let us now show that β satisfies $d(q, \lambda) \leq d(q, \beta) + 2^{i+1}$. Let $\gamma \in C \setminus N(q, \lambda) \cup \{\lambda\}$ be such that $\beta \in \mathcal{S}_i(\gamma, \mathcal{T}(R))$. Since $\gamma \notin N(q, \lambda) \setminus \{\lambda\}$, we get $d(\gamma, q) \geq d(q, \lambda)$. The triangle inequality says that $d(q, \gamma) \leq d(q, \beta) + d(\gamma, \beta)$. Finally Lemma D.13 implies that $d(\gamma, \beta) \leq 2^{i+1}$. Then

$$d(q, \lambda) \leq d(q, \gamma) \leq d(q, \beta) + d(\gamma, \beta) \leq d(q, \beta) + 2^{i+1}$$

So β is a desired k-nearest neighbor satisfying $d(q, \lambda) \leq d(q, \beta) + 2^{i+1}$. □

E. Construction of a compressed cover tree

This section introduces a new method Algorithm E.2 for construction of a compressed cover tree, which is based on Insert() method Beygelzimer et al. (2006a, Algorithm 2) that was specifically adapted for compressed cover tree. The proof of Beygelzimer et al. (2006a, Theorem 6), which estimated the time complexity of Beygelzimer et al. (2006a, Algorithm 2) was shown to be incorrect Elkin & Kurlin (2022a, Counterexample 4.2). The main contribution of this section are two new time complexity results that bound the time complexity of Algorithm E.2:

- Theorem 3.6 bounds the time complexity as $O(c_m(R)^{10} \cdot \log_2(\Delta(R)) \cdot |R|)$ by using minimized expansion constant $c_m(R)$ and aspect ratio $\Delta(R)$ as parameters.
- Theorem 3.9 bounds the time complexity as $O(c(R)^{12} \cdot \log_2 |R| \cdot |R|)$ by using expansion constant $c(R)$ as parameter.

Definition 2.10 explains the concrete implementation of compressed cover tree.

Definition 2.10 ($\mathrm{Children}(p, i)$ and $\mathrm{Next}(p, i, \mathcal{T}(R))$)**.** *In a compressed cover tree $\mathcal{T}(R)$ on a set R, for any level i and a node $p \in R$, set $\mathrm{Children}(p, i) = \{a \in \mathrm{Children}(p) \mid l(a) = i\}$. Let $\mathrm{Next}(p, i, \mathcal{T}(R))$ be the maximal level j satisfying $j < i$ and $\mathrm{Children}(p, i) \neq \emptyset$. If such level does not exist, we set $j = l_{\min}(\mathcal{T}(R)) - 1$. For every node p, we store its set of children in a linked hash map so that*

(a) *any key i gives access to* $\mathrm{Children}(p, i)$*,*

(b) $\mathrm{Children}(p, i) \to \mathrm{Children}(p, \mathrm{Next}(p, i, \mathcal{T}(R)))$*,*

(c) *we can directly access* $\max\{j \mid \mathrm{Children}(p, j) \neq \emptyset\}$*.*

Definition E.1 (construction iteration set $L(\mathcal{T}(W), p)$)**.** *Let W be a finite subset of a metric space (X, d). Let $\mathcal{T}(W)$ be a cover tree of Definition 2.1 built on W and let $p \in X \setminus W$ be an arbitrary point. Let $L(\mathcal{T}(W), p) \subseteq H(\mathcal{T}(R))$ be the set of all levels i during iterations 5-13 of Algorithm E.3 launched with inputs $\mathcal{T}(W), p$. Set $\eta(i) = \min_t\{t \in L(\mathcal{T}(W), p) \mid t > i\}$.*

Let R be a finite subset of a metric space (X, d). A compressed cover tree $\mathcal{T}(R)$ will be incrementally constructed by adding points one by one as summarized in Algorithm E.2. First we select a root node $r \in R$ and form a tree $\mathcal{T}(\{r\})$ of a

Algorithm E.2 Building a compressed cover tree $\mathcal{T}(R)$ from Definition 2.1.

```
1: Input : a finite subset R of a metric space (X, d)
2: Output : a compressed cover tree T(R).
3: Choose a random point r ∈ R to be a root of T(R)
4: Build the initial compressed cover tree T = T({r}) by making l(r) = +∞.
5: for p ∈ R \ {r} do
6:     T ← run AddPoint(T, p) described in Algorithm E.3.
7: end for
8: For root r of T set l(r) = 1 + max_{p∈R\{r}} l(p)
```

Algorithm E.3 Building $\mathcal{T}(W \cup \{p\})$ in lines 5-7 of Algorithm E.2.

```
Function AddPoint(a compressed cover tree T(W) with a root r, a point p ∈ X)
Output : compressed cover tree T(W ∪ {p}).
Set i ← l_max(T(W)) − 1 and η(l_max − 1) = l_max {If the root r has no children then i ← −∞}
Set R_{l_max} ← {r}.
while i ≥ l_min do
    Assign C_i(R_{η(i)}) ← R_{η(i)} ∪ {a ∈ Children(q) for some q ∈ R_{η(i)} | l(a) = i}
    Set R_i = {a ∈ C_i(R_{η(i)}) | d(p, a) ≤ 2^{i+1}}
    if R_i is empty then
        Launch Algorithm E.4 with parameters (p, R_{η(i)}).
    end if
    t = max_{a∈R_i} Next(a, i, T(W))  {If R_i has no children we set t = l_min − 1}
    η(i) ← i and i ← t
end while
Launch Algorithm E.4 with parameters (p, R_{η(i)}).
```

Algorithm E.4 Assign node subprocedure

```
Function AssignParent(Point p, subset of nodes U ⊆ T(W))
Output: Compressed cover tree T(W ∪ {p})
Pick v ∈ U minimizing d(v, p).
Set l(p) = ⌊log_2(d(p, v))⌋ − 1 and let v be a parent of p.
```

single node r at the level $l_{\max} = l_{\min} = +\infty$. Assume that we have a compressed cover tree $\mathcal{T}(W)$ for a subset $W \subset R$. For any point $p \in R \setminus W$, Algorithm E.3 builds a larger compressed cover tree $\mathcal{T}(W \cup \{p\})$ from $\mathcal{T}(W)$.

Note that during the construction of the compressed cover tree in Algorithm E.3 we write down additional information for every node p , which includes the number of descendants of node p and the maximal level of nodes in set $\mathrm{Children}(p)$.

Lemma E.5. *Let $\mathcal{T}(R)$ be a cover tree and let $p \in X$ be a point and let $i \in \mathbb{Z}$. Assume that for some $q \in \mathcal{T}(R)$ we have $d(p, q) > 2^{i+1}$. Let $\mathcal{S}_i(q, \mathcal{T}(R))$ be as defined in Definition 2.8. Then for any $\theta \in \mathcal{S}_i(q, \mathcal{T}(R)) \setminus \{q\}$ we have $d(\theta, p) > 2^{l(\theta)}$.*

Proof. Let $S = (\theta = a_0, ..., a_m = q)$ be a node to node path. Since $\theta \in \mathcal{S}_i(q, \mathcal{T}(R)) \setminus \{q\}$ by Lemma 2.9 we have $l(a_{m-1}) \leq i - 1$. Therefore $l(\theta) = l(a_0) \leq ... \leq l(a_{m-1}) \leq i - 1$. We have the following inequality:

$$d(q, \theta) \leq \sum_{z=0}^{h-1} d(a_z, a_{z+1}) \leq \sum_{x=l(\theta)+1}^{j} 2^x = (2^{j+1} - 2^{l(\theta)+1}).$$

By triangle inequality we have: $d(p, \theta) \geq d(p, \gamma) - d(\gamma, \theta) > 2^{j+1} - (2^{j+1} - 2^{l(\theta)+1}) > 2^{l(\theta)}$. Therefore $d(p, \theta) > 2^{l(\theta)}$ which proves the claim. □

Theorem 3.2 (correctness of Algorithm 3.4)**.** *Algorithm 3.4 builds a compressed cover tree in Definition 2.1.*

Proof. It suffices to prove that Algorithm E.3 correctly extends a compressed cover tree $\mathcal{T}(W)$ for any finite subset $W \subseteq X$ by adding a point p. Let us prove that $\mathcal{T}(W \cup \{p\})$ satisfies Definition 2.1.

We first note that the parent v of p is always assigned in Algorithm E.4 by setting $l(p) = \lfloor \log_2(d(p, v) \rfloor - 1$. Note that the set U is never empty, when Algorithm E.4 is launched. The covering condition (2.1b) after adding point p to $\mathcal{T}(W)$ follows from the following inequality:

$$d(p, v) \leq 2^{\lfloor \log_2(d(p,v) \rfloor} \leq 2^{l(p)+1}.$$

To check (2.1c) Consider arbitrary cover set $C_h = \{q \in \mathcal{T}(W \cup \{p\}) \mid l(q) \geq h\}$. Since we have assumed that $\mathcal{T}(W)$ is a valid cover tree, all the cover sets C_h for $h > l(p)$ satisfy the condition. Let us consider cover sets having $h \leq l(p)$. Let $\theta \in C_h$ be an arbitrary node. Consider a sequence of iterations $l_{\min}(\mathcal{T}(W)) \leq a(0) < a(1) < ... < a(t) = l_{\max}(\mathcal{T}(W))$ that were considered during run-time of the algorithm. Note that the parent of p was assigned at $i = a(0)$. Since $\theta \in W = \mathcal{S}_{l_{\max}(r, \mathcal{T}(W))}$, either (**a**) $\theta \in \bigcup_{q \in R_{a(0)}} \mathcal{S}_{a(0)}(q, R_{a(0)})$ or (**b**) there exists index j satisfying

$$\theta \in \bigcup_{q \in R_{a(j+1)}} \mathcal{S}_{a(j+1)}(q, \mathcal{T}(W)) \setminus \bigcup_{q \in R_i} \mathcal{S}_{a(j)}(q, \mathcal{T}(W)).$$

Let us first consider case (**a**). Let v be a parent of p in $\mathcal{T}(W \cup \{p\})$. Recall that the parent v of p was assigned in line 4 of Algorithm E.4. Therefore we have $d(v, p) \leq d(p, \theta)$ and by line 4 we have:

$$d(p, \theta) \geq d(p, v) \geq 2^{l(p)+1} > 2^{l(p)} \geq 2^h,$$

which proves the claim.

Assume now (**b**) holds. Denote $i = a(j + 1)$, since $a(j)$ was previous level, it follows $\eta(i) = a(j)$. By Lemma D.14 we have:

$$\bigcup_{q \in \mathcal{C}_i(R_{\eta(i)})} \mathcal{S}_i(q, \mathcal{T}(W)) = \bigcup_{q \in R_{\eta(i)}} \mathcal{S}_{\eta(i)}(q, \mathcal{T}(W)).$$

Therefore there exists a node $u \in \mathcal{C}_i(R_{\eta(i)}) \setminus R_i$ for which $\theta \in \mathcal{S}_i(u, \mathcal{T}(W))$. By line 7 of Algorithm E.3 we have $d(u, p) > 2^{i+1}$. If $u = \theta$, then the parent of p was selected from set $R_{\eta(i)}$ and the proof is similar to (**a**). Else by Lemma E.5 it follows that $d(p, \theta) > 2^{l(\theta)} \geq 2^h$ which proves the claim.

□

Lemma 3.3 (time complexity of a key step for $\mathcal{T}(R)$)**.** *Arbitrarily order all points of a finite reference set R in a metric space (X,d) starting from the root: $r = p_1, p_2, \dots, p_{|R|}$. Set $W_1 = \{r\}$ and $W_{y+1} = W_y \cup \{p_y\}$ for $y = 1, ..., |R|-1$. Then Algorithm 3.4 builds a compressed cover tree $\mathcal{T}(R)$ in time*

$$O\Big((c_m(R))^8 \cdot \max_{y=1,\dots,|R|-1} L(\mathcal{T}(W_y), p_y) \cdot |R|\Big),$$

where $c_m(R)$ is the minimized expansion constant from Definition 1.4.

Proof. The worst-case time complexity of Algorithm E.2 is dominated by lines 5-7 which call Algorithm E.3 $O(|R|)$ times in total.

Assume that we have already constructed a cover tree on set $\mathcal{T}(W_y)$, the goal Algorithm E.3 is to construct tree $\mathcal{T}(W_y \cup \{p_{y+1}\})$. By Definition E.1 loop on lines 5-13 is performed $L(\mathcal{T}(W_y), p_{y+1})$ times. Let R_* be the maximal size of set R_i during all iterations $i \in L(\mathcal{T}(W_y), p_{y+1})$. By Lemma 2.3 since $W_{y+1} \subseteq R \subseteq X$ we have

$$|\mathcal{C}_i(R_{\eta(i)})| \leq c_m(W_{y+1})^4 \cdot |R_*| \leq c_m(R)^4 \cdot |R_*|$$

nodes, where $\mathcal{C}_{\eta(i)}(R_{\eta(i)})$ is defined in line 6. Therefore both, lines 7 and 6 take at most $c_m(R)^4|R_*|$ time. In line 11 we handle $|R_*|$ elements, for each of them we can retrieve index $\mathrm{Next}(a, i, \mathcal{T}(W_y))$ in $O(1)$ time, since for every $a \in \mathcal{T}(R)$ we can update the last index j, when a had children on level j in line 6. Therefore line 11 takes at most $O(|R_*|)$ time. Algorithm E.4 takes at most $O(|R_*|)$ time. Therefore line 9 and line 14 take at most $O(|R_*|)$ time. Let us now bound $|R_*|$ during the whole run-time of the algorithm.

Let i be an arbitrary level. Note that $R_i \subseteq B(p, 2^{i+1}) \cap C_i$ where C_i is a ith cover set of $\mathcal{T}(R)$. Since C_i is 2^i-spares subset of R we can apply packing Lemma 2.2 with $r = 2^{i+1}$ and $\delta = 2^i$ to obtain $|B(p, 2^{i+1}) \cap C_i| \leq (c_m(W))^4$. Lemma 1.5 implies $(c_m(W))^4 \leq (c_m(R))^4$, therefore $|B(p, 2^i) \cap C_i| \leq (c_m(R))^4$.

The time complexity of loop 5 - 13 in Algorithm E.3 is dominated by line 6 that has time $O(|C(R_i)|) \leq O((c_m(R))^4 \cdot |R_*|) \leq O((c_m(R))^8)$. Then the whole Algorithm E.2 has time

$$O((c_m(R))^8 \cdot \max_{y=2,\dots,|R|} L(\mathcal{T}(W_{y-1}), p_y) \cdot |R|)$$

as desired. □

Theorem 3.6 (time complexity of $\mathcal{T}(R)$ via aspect ratio)**.** *Let R be a finite subset of a metric space (X,d) having the aspect ratio $\Delta(R)$. Algorithm 3.4 builds a compressed cover tree $\mathcal{T}(R)$ in time $O((c_m(R))^8 \cdot \log_2(\Delta(R)) \cdot |R|)$, where $c_m(R)$ is the minimized expansion constant from Definition 1.4.*

Proof. In Lemma 3.3 use the upper bounds due to Lemma B.8 as follows:
$\max_{y \in 2,\dots,|R|} |L(\mathcal{T}(W_{y-1}), p_y)| \leq H(\mathcal{T}(R)) \leq 1 + \log_2(\Delta(R))$. □

Lemma 3.7. *Let (X,d) be a metric space and let $W \subseteq X$ be its finite subset. Let $q \in X \setminus W$ be an arbitrary point. Let $i \in L(\mathcal{T}(W), q)$ be arbitrarily iteration of Definition 3.1. Assume that $t = \eta(\eta(i+1))$ is defined. Then there exists $p \in W$ satisfying $2^{i+1} < d(p,q) \leq 2^{t+1}$.*

Proof. Note first that since $\eta(i+3) \in L(\mathcal{T}(R), q)$, there exists distinct $u \in R_{\eta(\eta(i+3))}$ and $v \in \mathcal{C}_{\eta(i+1)}(R_{\eta(\eta(i+1))})$, in such a way that u is the parent of v. Let us show that both of u, v cant belong to set R_i. Assume contrary that both $u, v \in R_i$. Then by line 7 of Algorithm E.3 we have $d(v,q) \leq 2^{i+1}$ and $d(u,q) \leq 2^{i+1}$. By triangle inequality $d(u,v) \leq d(u,q) + d(q,v) \leq 2^{i+2} \leq 2^{\eta(i+1)}$. Recall that we denote a level of a node by l. On the other hand we have $l(u) \geq \eta(i+1)$ and $l(v) \geq \eta(i+1)$, by separation condition of Definition 2.1 we have $d(u,v) > 2^{\eta(i+1)}$, which is a contradiction. Therefore only one of $\{u, v\}$ can belong to R_i. It sufficies two consider the two cases below:

Assume that $v \notin R_i$. Since v is children of u we have $d(u,v) \leq 2^{\eta(i+1)+1}$. By line 7 of Algorithm E.3 we have $d(u,q) \leq 2^{\eta(i+1)+1}$. By triangle inequality

$$d(v,q) \leq d(v,u) + d(u,q) \leq 2^{\eta(i+1)+1} + 2^{\eta(i+1)+1} \leq 2^{\eta(i+1)+2} \leq 2^{\eta(\eta(i+1))+1}$$

Since $v \notin R_i$ there exists level t having $\eta(i+1) \geq t \geq i$ and $v \in \mathcal{C}_t(R_{\eta(t)}) \setminus R_t$. Therefore by line 7 of Algorithm E.3 we have $d(q,v) > 2^{t+1} \geq 2^{i+1}$. It follows that we have found point $v \in R$ satisfying $2^{i+1} < v \leq 2^{\eta(\eta(i+1))+1}$. Therefore $p = v$, is the desired point.

Assume that $u \notin R_i$. Since $u \in R_{\eta(\eta(i+1))}$, by line 7 of Algorithm E.3 we have $d(u,q) \leq 2^{\eta(\eta(i+1))+1}$. On the other hand since $u \notin R_i$, there exists level t having $\eta(i+3) \geq t \geq i$ and $u \in \mathcal{C}_t(R_{\eta(t)}) \setminus R_t$. Therefore by line 7 of Algorithm E.3 we have $d(q,u) > 2^{t+2} \geq 2^{i+2}$. It follows that we have found point $u \in R$ satisfying $2^{i+1} < u \leq 2^{\eta(\eta(i+1))+1}$. Therefore $p = u$, is the desired point.

□

Lemma 3.8 (Construction iteration bound)**.** *Let A, W be finite subsets of a metric space X satisfying $W \subseteq A \subseteq X$. Take a point $q \in A \setminus W$. Given a compressed cover tree $\mathcal{T}(W)$ on W, Algorithm 3.5 runs lines 5-14 this number of times: $|L(\mathcal{T}(W), q)| = O\big(c(A)^2 \cdot \log_2(|A|)\big)$.*

Proof. Let $x \in L(\mathcal{T}(R), q)$ be the lowest level of $L(\mathcal{T}(R), q)$. Define $s_1 = \eta(\eta(x)+1)$ and let $s_i = \eta(\eta(\eta(s_{i-1}+1))+1)$, if it exists. Assume that s_{n+1} is the last sequence element for which $\eta(\eta(\eta(s_{n-1}+1))+1)$ is defined. Define $S = \{s_1, ..., s_n\}$. For every $i \in \{1, ..., n\}$ let p_i be the point provided by Lemma 3.7 that satisfies

$$2^{s_i+1} < d(p_i, q) \leq 2^{\eta(\eta(s_i+1))+1}.$$

Let P be the sequence of points p_i. Denote $n = |P| = |S|$. Let us show that S satisfies the conditions of Lemma 2.5. Note that:

$$4 \cdot d(p_i, q) \leq 4 \cdot 2^{\eta(\eta(s_i+1))+1} \leq 2^{\eta(\eta(s_i+1))+3} \leq 2^{\eta(\eta(\eta(s_i+1))+1)+1} \leq 2^{s_{i+1}+1} \leq d(p_{i+1}, q)$$

By Lemma 2.5 applied for set A and sequence P we get:

$$|\bar{B}(q, \frac{4}{3}d(q,p_n))| \geq (1 + \frac{1}{c(R)^2})^n \cdot |\bar{B}(q, \frac{1}{3}d(q,p_1))|$$

Since $\eta(x) \in L(\mathcal{T}(R), q)$, there exists some point $u \in R_{\eta(x)}$. By definition of R_i we have $d(u,q) \leq 2^{\eta(x)+1}$. Also

$$2^{\eta(\eta(x)+1)-1} \leq \frac{2^{\eta(\eta(x)+1)+1}}{3} < \frac{d(q,p_1)}{3}$$

It follows that:

$$1 \leq |\bar{B}(q, 2^{\eta(x)+1})| \leq |\bar{B}(q, 2^{\eta(\eta(x)+1)-1}| \leq |\bar{B}(q, \frac{d(q,p_1)}{3})|$$

Therefore we have

$$|A| \geq \frac{|\bar{B}(q, \frac{4}{3} \cdot d(q,p_n))|}{|\bar{B}(q, \frac{1}{3} \cdot d(q,p_1))|} \geq (1 + \frac{1}{c(A)^2})^n$$

Note that $c(A) \geq 2$ by definition of expansion constant. Then by applying $\log$ and by using Lemma B.7 we obtain: $c(A)^2 \log(A) \geq n = |S|$. Let x be minimal level of $L(\mathcal{T}(W), q)$ and let y be the maximal level of $L(\mathcal{T}(W), q)$ Note that S is a sub sequence of L in such a way that:

- $[x, s_1] \cap L(\mathcal{T}(R), q) \leq 3$,
- for all $i \in 1, ..., n$ we have $[s_i, s_{i+1}] \cap L(\mathcal{T}(R), q) \leq 6$
- $[s_n, y] \cap L(\mathcal{T}(R), q) < 12$

Since segments $[x, s_1], [s_1, s_2], ..., [s_2, s_n], [s_n, y]$ cover $|L(\mathcal{T}(R), q)|$, it follows that $|S| \geq \frac{|L(\mathcal{T}(R),q)|}{12}$. We obtain that

$$|L(\mathcal{T}(R), q)| \leq 12 \cdot c(A)^2 \cdot \log_2(|A|),$$

which proves the claim. □

Theorem 3.9 (time for $\mathcal{T}(R)$ via expansion constants)**.** *Let R be a finite subset of a metric space (X, d). Let A be a finite subset of X satisfying $R \subseteq A \subseteq X$. Then Algorithm 3.4 builds a compressed cover tree $\mathcal{T}(R)$ in time $O((c_m(R))^8 \cdot c(A)^2 \cdot \log_2(|A|) \cdot |R|)$, see the expansion constants $c(A), c_m(R)$ in Definition 1.4.*

Proof. It follows from Lemmas 3.8 and 3.3. □

Corollary 3.10. *Let R be a finite subset of a metric space (X, d). Then Algorithm 3.4 builds a compressed cover tree $\mathcal{T}(R)$ in time $O((c_m(R))^8 \cdot c(R)^2 \cdot \log_2(|R|)) \cdot |R|)$, where the constants $c(R), c_m(R)$ appeared in Definition 1.4.*

Proof. The proof follows from Theorem 3.9 by setting $A = R$. □

F. k-nearest neighbor search algorithm

This section is motivated by Elkin & Kurlin (2022a, Counterexample 5.2), which showed that the proof of past time complexity claim in Beygelzimer et al. (2006a, Theorem 5) for the nearest neighbor search algorithm contained gaps. The two main results of this sections are Corollary 4.7 and Theorem 4.9 which provide new time complexity results for k-nearest neighbor problem, assuming that a compressed cover tree was already constructed for the reference set R. For the construction algorithm of compressed cover tree and its time complexity, we refer to Section E.

The past mistakes are resolved by introducing a new Algorithm F.2 for finding k-nearest neighbors that generalize and improves the original method for finding nearest neighbors using an implicit cover. Beygelzimer et al. (2006a, Algorithm 1). The first improvement is λ-point of line 6 which allows us to search for all k-nearest neighbors of a given query point for any $k \geq 1$. The second improvement is a new loop break condition on line 8. The new loop break condition is utilized in the proof of Lemma 4.8 to conclude that the total number of performed iterations is bounded by $O(c(R)^2 \log(|R|))$ during whole run-time of the algorithm. The latter improvement closes the past gap in proof of Beygelzimer et al. (2006a, Theorem 5) by bounding the number of iterations independently from the explicit depth Elkin & Kurlin (2022a, Definition 3.2), that generated the past confusion.

Recall from Definition D.2 that an essential set $\mathcal{E}(p, \mathcal{T}(R)) \subseteq H(\mathcal{T}(R)$ consists of all levels $i \in H(\mathcal{T}(R))$ for which p has non-trivial children in $\mathcal{T}(R)$ at level i. By Lemma D.5 the sizes of distinctive descendants $|\mathcal{S}_i(p, \mathcal{T}(R))|$ can be precomputed in a linear time $O(|R|)$ for all $p \in R$ and $i \in \mathcal{E}(p, \mathcal{T}(R))$. Since the size of distinctive descendant set $|\mathcal{S}_i(p, \mathcal{T}(R))|$ can only change at indices $i \in \mathcal{E}(p, \mathcal{T}(R))$, we assume that the sizes of $|\mathcal{S}_i(p, \mathcal{T}(R))|$ can be retrieved in a constant time $O(1)$ for any $p \in R$ and $i \in H(\mathcal{T}(R))$ during the run-time of Algorithm F.2.

Definition F.1. *Let R be a finite subset of a metric space (X, d). Let $\mathcal{T}(R)$ be a cover tree of Definition 2.1 built on R and let $q \in X$ be arbitrary point. Let $L(\mathcal{T}(R), q) \subseteq H(\mathcal{T}(R))$ be the set of all levels i during iterations of lines 4-17 of Algorithm F.2 launched with inputs $\mathcal{T}(R), q$. If Algorithm F.2 reaches line 13 at level $\varrho \in L(\mathcal{T}(R), q)$, then we say that is* special. *We denote $\eta(i) = \min_t\{t \in L(\mathcal{T}(R), q) \mid t > i\}$.* ■

Note that $\eta(i)$ of Definition F.1 may be undefined. If $\eta(i)$ is defined, then by definition we have $\eta(i) \geq i + 1$. Let $d_k(q, R)$ be the distance of q to its kth nearest neighbor in R.

Example F.4 (Simulated run of Algorithm F.2)**.** Let R and $\mathcal{T}(R)$ be as in Example B.2. Let $q = 0$ and $k = 5$. Figures 10, 11, 12 and 13 illustrate simulated run of Algorithm F.2 on input $(\mathcal{T}(R), q, k)$. Recall that $l_{\max} = 2$ and $l_{\min} = -1$. During the iteration i of Algorithm F.2 we maintain the following coloring: Points in R_i are colored orange. Points $\mathcal{C}_{\eta(i)}(R_{\eta(i)})$ (of line 5) that are not contained in R_i are colored yellow. The λ-point of line 6 is denoted by using purple color. All the nodes that were present in $R_{\eta(i)}$, but are no longer included in R_i will be colored red. Finally all the points that are selected as k-nearest neighbors of q are colored green in the final iteration. Nodes that haven't been yet visited or that will never be visited are colored white. Let $R_2 = \{8\}$. Consider the following steps:

Iteration $i = 1$: Figure 10 illustrates iteration $i = 1$ of the Algorithm F.2. In line 5 we find $\mathcal{C}_1(R_2) = \{4, 8, 12\}$. Since node 4 minimizes distance $d(\mathcal{C}_1(R_2), 0)$ and distinctive descendant set $\mathcal{S}_2(4, \mathcal{T}(R))$ consists of 7 elements we get $\lambda = 4$ and therefore $d(q, \lambda) = 4 \leq 2^{i+2} = 8$. In line 7 we find $R_1 = \{r \in C \mid d(0, r) \leq d(q, \lambda) + 2^3 = 12\} = \{4, 8, 12\}$.

Iteration $i = 0$: Figure 11 illustrates iteration $i = 0$ of the Algorithm F.2. In line 5 we find $\mathcal{C}_0(R_1) = \{2, 4, 6, 8, 10, 12, 14\}$. Since $|\mathcal{S}_1(2, \mathcal{T}(R))| = 3$, $|\mathcal{S}_1(4, \mathcal{T}(R))| = 1$ and $|\mathcal{T}_1(6)| = 3$ and 6 is the node with smallest to distance 0 satisfying

Algorithm F.2 k-nearest neighbor search by a compressed cover tree

1: **Input** : compressed cover tree $\mathcal{T}(R)$, a query point $q \in X$, an integer $k \in \mathbb{Z}_+$
2: Set $i \leftarrow l_{\max}(\mathcal{T}(R)) - 1$ and $\eta(l_{\max} - 1) = l_{\max}$
3: Let r be the root node of $\mathcal{T}(R)$. Set $R_{l_{\max}} = \{r\}$.
4: **while** $i \geq l_{\min}$ **do**
5: Assign $\mathcal{C}_i(R_{\eta(i)}) \leftarrow R_{\eta(i)} \cup \{a \in \text{Children}(p) \text{ for some } p \in R_{\eta(i)} \mid l(a) = i\}$ $\{$Recall that $\text{Children}(p)$ contains node p $\}$
6: Compute $\lambda = \lambda_k(q, \mathcal{C}_i(R_{\eta(i)}))$ from Definition D.6 by Algorithm D.8.
7: Find $R_i = \{p \in \mathcal{C}_i(R_{\eta(i)}) \mid d(q,p) \leq d(q,\lambda) + 2^{i+2}\}$
8: **if** $d(q,\lambda) > 2^{i+2}$ **then**
9: Define list $S = \emptyset$
10: **for** $p \in R_i$ **do**
11: Update S by running Algorithm F.3 on (p,i)
12: **end for**
13: Compute and **output** k-nearest neighbors of the query point q from set S.
14: **end if**
15: Set $j \leftarrow \max_{a \in R_i} \text{Next}(a, i, \mathcal{T}(R))$ $\{$If such j is undefined, we set $j = l_{\min} - 1\}$
16: Set $\eta(j) \leftarrow i$ and $i \leftarrow j$.
17: **end while**
18: Compute and **output** k-nearest neighbors of query point q from the set $R_{l_{\min}}$.

Algorithm F.3 The node collector called in line 11 of Algorithm F.2.

1: **Input:** $p \in R$, index i.
2: **Output:** a list $S \subseteq R$ containing all nodes of $\mathcal{S}_i(p, \mathcal{T}(R))$.
3: Add p to list S.
4: **if** $i > l_{\min}(\mathcal{T}(R))$ **then**
5: Set $j = \text{Next}(p, i, \mathcal{T}(R))$
6: Set $C = \{a \in \text{Children}(p) \mid l(a) = j\}$
7: **for** $u \in C$ **do**
8: Call Algorithm F.3 with (u, j).
9: **end for**
10: **end if**

$\sum_{p \in N(0,6)=\{2,4,6\}} |\mathcal{S}_1(p, \mathcal{T}(R))| \geq 5 = k$. It follows that $\lambda = 6$. In line 7 we find $R_0 = \{r \in \mathcal{C}(R_1) \mid d(0,r) \leq d(q,\lambda) + 2^2 = 10\} = \{2,4,6,8,10\}$. Since $d(q,\lambda) > 2^{i+2} = 4$. We proceed into lines 8 - 14

Final block lines 8 - 14 for $i = 0$: Figure 12 marks all the points S discovered by line 11 as orange. Figure 13 illustrates the final selection of k points from set S that are selected as the final output $\{1,2,3,4,5\}$.

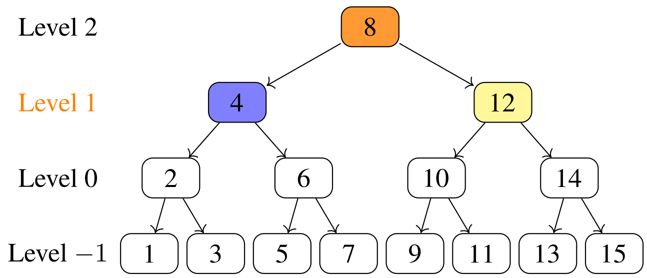

Figure 10. Iteration $i = 1$ of simulation in Example F.4 of Algorithm F.2

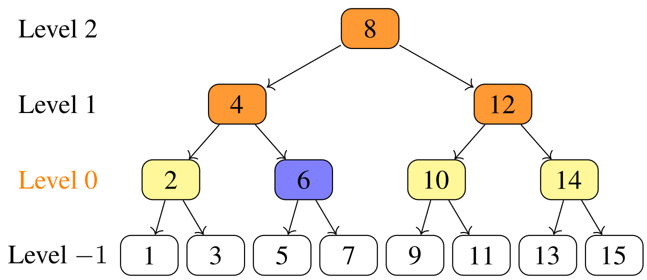

Figure 11. Iteration $i = 0$ of simulation in Example F.4 of Algorithm F.2

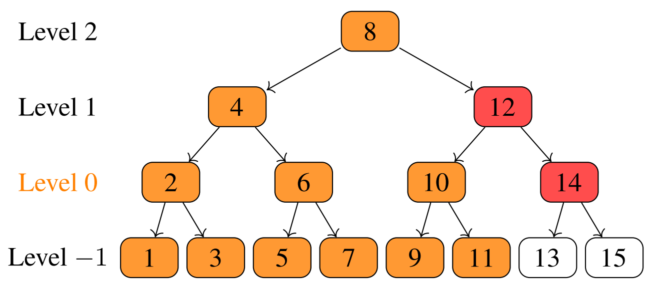

Figure 12. Line 11 of Iteration $i = 0$ of simulation in Example F.4 of Algorithm F.2

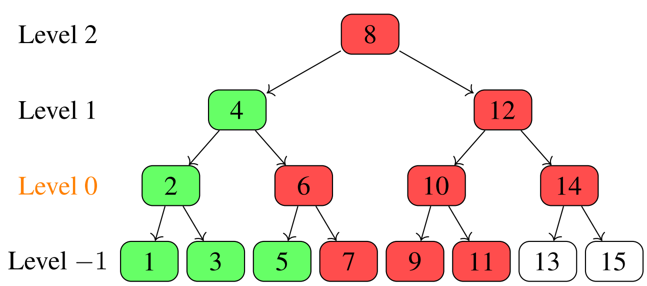

Figure 13. Line 13 of iteration $i = 0$ of simulation in Example F.4 of Algorithm F.2

Note that $\bigcup_{p\in R_i} \mathcal{S}_i(p,\mathcal{T}(R))$ is decreasing set for which $\bigcup_{p\in R_{l_{\max}}} \mathcal{S}_{l_{\max}}(p,\mathcal{T}(R)) = R$ and

$$\bigcup_{p\in R_{l_{\min}}} \mathcal{S}_{l_{\min}}(p,\mathcal{T}(R)) = R_{l_{\min}}.$$

Lemma F.5 (k-nearest neighbors in the candidate set for all i)**.** *Let R be a finite subset of an ambient metric space (X,d), let $q\in X$ be a query point and let $k\in\mathbb{Z}\cap[1,\infty)$ be a parameter. Let $\mathcal{T}(R)$ be a compressed cover tree of R. Assume that $|R|\geq k$. Then for any iteration $i\in L(\mathcal{T}(R),q)$ of Definition F.1 the candidate set $\bigcup_{p\in R_i}\mathcal{S}_i(p,\mathcal{T}(R))$ contains all k-nearest neighbors of q.* ■

Proof. Since $R_{l_{\max}} = \{r\}$, where r is the root $\mathcal{T}(R)$ we have $S_{l_{\max}}(r,\mathcal{T}(R)) = R$ and therefore any point among k-nearest neighbor of q is contained in $R_{l_{\max}}$. Let i be the largest index for which there exists a point among k-nearest neighbor of q that doesn't belong to $\bigcup_{p\in R_i}\mathcal{S}_i(p,\mathcal{T}(R))$. Let us denote such point by β, then:

$$\beta\in\bigcup_{p\in R_{\eta(i)}}\mathcal{S}_{\eta(i)}(p,\mathcal{T}(R))\setminus\bigcup_{p\in R_i}\mathcal{S}_i(p,\mathcal{T}(R)).$$

By Lemma D.14 we have

$$\bigcup_{p\in\mathcal{C}_{\eta(i)}(R_{\eta(i)})}\mathcal{S}_i(p,\mathcal{T}(R)) = \bigcup_{p\in R_{\eta(i)}}\mathcal{S}_{\eta(i)}(p,\mathcal{T}(R)) \tag{6}$$

Let λ be as in line 6 of Algorithm F.2. By Equation (6) we have

$$|\bigcup_{p\in\mathcal{C}_{\eta(i)}(R_{\eta(i)})}\mathcal{S}_i(p,\mathcal{T}(R))|\geq k,$$

therefore by Definition D.6 such λ exists. Since $\beta\in\bigcup_{p\in\mathcal{C}_{\eta(i)}(R_{\eta(i)})}\mathcal{S}_i(p,\mathcal{T}(R))$, there exists $\alpha\in\mathcal{C}_{\eta(i)}(R_{\eta(i)})$ satisfying $\beta\in\mathcal{S}_i(\alpha,\mathcal{T}(R))$. By assumption it follows $\alpha\notin R_i$. By line 7 of the algorithm we have

$$d(\alpha,q) > d(q,\lambda)+2^{i+2}. \tag{7}$$

Let w be arbitrary point in set $\bigcup_{p\in N(q;\lambda)}\mathcal{S}_i(p,\mathcal{T}(R))$. Therefore $w\in\mathcal{S}_i(\gamma,\mathcal{T}(R))$ for some $\gamma\in N(q;\lambda)$. By Lemma D.13 applied on i we have $d(\gamma,w)\leq 2^{i+1}$. By Definition D.6 since $\gamma\in N(q;\lambda)$ we have $d(q,\gamma)\leq d(q,\lambda)$. By (7) and the triangle inequality we obtain:

$$d(q,w)\leq d(q,\gamma)+d(\gamma,w)\leq d(q,\lambda)+2^{i+1} < d(\alpha,q)-2^{i+1} \tag{8}$$

On the other hand β is a descendant of α thus we can estimate:

$$d(q,\beta)\geq d(q,\alpha)-d(\alpha,\beta)\geq d(\alpha,q)-2^{i+1} \tag{9}$$

By combining Inequality (8) with Inequality (9) we obtain $d(q,w) < d(q,\beta)$. Since w was arbitrary point from $\bigcup_{p\in N(q;\lambda)}\mathcal{S}_i(p,\mathcal{T}(R))$, that contains at least k points, β cannot be any k-nearest neighbor of q, which is a contradiction. □

Theorem 4.4 (correctness of Algorithm 4.3)**.** *Algorithm 4.3 correctly finds all k-nearest neighbors of query point q within the reference set R.*

Proof. Note that Algorithm F.2 is terminated by either reaching line 18 or by going inside block 10 - 12.

Assume first that Algorithm F.2 is terminated by reaching line 18. Claim follows directly from Lemma F.5 by noting that since $i = l_{\min}$ all the nodes $p\in R_{l_{\min}}$ do not have any children. Therefore it follows $\bigcup_{p\in R_{l_{\min}}}\mathcal{S}_i(p,\mathcal{T}(R)) = R_{l_{\min}}$. Thus all the k-nearest neighbors of q are contained in the set $R_{l_{\min}}$.

Assume then that block 10 - 12 is reached during some iteration $i\in L(\mathcal{T}(R),q)$. By Lemma F.5 set $\bigcup_{p\in R_i}\mathcal{S}_i(p,\mathcal{T}(R))$ contains all k-nearest neighbors of q. Note that in line 11 we collect all nodes of $\bigcup_{p\in R_i}\mathcal{S}_i(p,\mathcal{T}(R))$ into single array S. Therefore in line 13 we correctly select k nearest neighbors of q from array S, which proves the claim. □

Lemma 4.5. *Algorithm 4.3 has the following time complexities of its lines*

(a) $\max\{\#\text{Line}[4-9], \#\text{Line}[12-15], \#\text{Line}[16]\} = O\big(c_m(R)^{10} \cdot \log_2(k)\big)$*;*

(b) $\#\text{Line}[8-14] = O\big(|\bar{B}(q, 5d_k(q,R))| \cdot \log_2(k)\big)$.

Proof. **(a)** Let $\varrho \in L(\mathcal{T}(R), q)$ be as in Definition F.1. Note that if iteration ϱ is encountered, it becomes the last iteration of $L(\mathcal{T}(R), q)$. The total number of children encountered in line 5 during single iteration (4-17) is at most is at most $(c_m(R))^4 \cdot \max\limits_{i \in L(\mathcal{T}(R),q) \setminus \varrho} |R_i|$ by Lemma 2.3. From Lemma D.10 we obtain that line 6, which launches Algorithm D.8 takes at most

$$|\mathcal{C}(R_i)| \cdot \log_2(k) = (c_m(R))^4 \cdot \max_{L(q,\mathcal{T}(R)) \setminus \varrho} |R_i| \cdot \log_2(k)$$

time. Line 7 never does more work than line 5, since in the worst case scenario $R_{\eta(i)}$ is copied to R_i in its current form. Line 15 handles $|R_i|$ nodes, since we can keep track of value of $\text{Next}(a, i, \mathcal{T}(R))$ of Definition 2.10 by updating it when necessary in line 5 we can retrieve its value in $O(1)$ time. Therefore maximal run-time of line 15 is $\max\limits_{i \in L(q,\mathcal{T}(R)) \setminus \varrho} |R_i|$. Final line 18 picks lowest k-elements $R_{\eta(i)}$ ranked by function $f(p) = d(p,q)$. By Lemma D.9 it can be computed in time $O(\log_2(k) \cdot \max\limits_{L(q,\mathcal{T}(R)) \setminus \varrho} |R_i|)$. It follows that

$$\max(\#\text{Line}[4,8], \#\text{Line}[14,17], \#\text{Line}[18]) = O\Big(c_m(R)^4 \cdot \max_{i \in L(q,\mathcal{T}(R)) \setminus \varrho} |R_i| \cdot \log_2(k)\Big) \tag{10}$$

Let us now bound $\max_{i \in L(q,\mathcal{T}(R)) \setminus \varrho} |R_i|$, by showing $|R_i| \le c_m(R)^6$. Let C_i be the ith level of $\mathcal{T}(R)$ as in Definition 2.1. For all $i \in L(\mathcal{T}(R), q) \setminus \varrho$ we have:

$$R_i = \{r \in \mathcal{C}_i(R_{\eta(i)}) \mid d(p,q) \le d(q,\lambda) + 2^{i+2}\} \tag{11}$$

$$= B(q, d(q,\lambda) + 2^{i+2}) \cap \mathcal{C}_i(R_i) \tag{12}$$

$$\subseteq B(q, 2^{i+3}) \cap C_i \tag{13}$$

From cover-tree condition we know that all the points in C_i are separated by 2^i. We will now apply Lemma 2.2 with $t = 2^{i+3}$ and $\delta = 2^i$. Since $4\frac{t}{\delta} + 1 = 2^5 + 1 \le 2^6$ we obtain $\max\limits_{i \in L(q,\mathcal{T}(R)) \setminus \varrho} |R_i| \le |B(q, 2^{i+2}) \cap C_i| \le c_m(R)^6$. The claim follows by replacing $\max\limits_{i \in L(q,\mathcal{T}(R)) \setminus \varrho} |R_i|$ with $c_m(R)^6$ in (10).

(b) Let us now bound the run-time of $\#\text{Line}[8,17]$. which runs Algorithm F.3 for all (p,i), where $p \in R_i$. Let $\mathcal{S}$ be a distinctive descendant set from Definition 2.8. Algorithm F.3 visits every node $u \in \cup_{p \in R_i} \mathcal{S}_i(p, \mathcal{T}(R))$ once, therefore its running time is $O(\cup_{p \in R_i} |\mathcal{S}_i(p, \mathcal{T}(R))|)$. Let us now show that

$$\cup_{p \in R_i} \mathcal{S}_i(p, \mathcal{T}(R)) \subseteq \bar{B}(q, 5d_k(q,R))$$

Note first that by Lemma F.5 set $\cup_{p \in R_i} \mathcal{S}_i(p, \mathcal{T}(R))$ contains all k-nearest neighbors of q. Using Lemma D.15 we find β among k-nearest neighbors of q satisfying $d(q,\lambda) \le d(q,\beta) + 2^{i+1}$. From assumption It follows $2^{i+1} \le d(q,\beta)$.

By line 8 we have $d(q,\lambda) \le 2^{i+1}$. By line 13 we perform depth-first traversal on

$$A = \cup_{p \in R_i} \mathcal{S}_i(p, \mathcal{T}(R)).$$

Let $u \in \cup_{p \in R_i} \mathcal{S}_i(p, \mathcal{T}(R))$ be arbitrary node and let $v \in R_i$ be such that $u \in \mathcal{S}_i(v, \mathcal{T}(R))$. By Lemma D.13 we have $d(u,v) \le 2^{i+1}$. Since $v \in R_i$ we have $d(q,v) \le d(\lambda,q) + 2^{i+2}$. By triangle inequality

$$d(u,q) \le d(u,v) + d(v,q) \le 2^{i+1} + d(\lambda,v) + 2^{i+2} \le 2^{i+1} + 2^{i+1} + d(q,\beta) + 2^{i+2} \le 5 \cdot d(q,\beta)$$

It follows that $\cup_{p \in R_i} \mathcal{S}_i(p, \mathcal{T}(R)) \subseteq \bar{B}(q, 5 \cdot d(q,\beta))$. Let us now bound the time complexity of line 13. By Lemma D.9 for any set A is takes $\log(k) \cdot |A|$ time to select k-lowest elements. We have:

$$\#\text{Line}[8,17] = O(|\bar{B}(q, 5 \cdot d_k(q,R))| \cdot \log(k)).$$

□

Theorem 4.6. *Let R be a finite set in a metric space (X, d), $c_m(R)$ be the minimized constant from Definition 1.4. Given a compressed cover tree $\mathcal{T}(R)$, Algorithm 4.3 finds all k-nearest neighbors of a query point $q \in X$ in time*

$$O\Big(\log_2(k) \cdot ((c_m(R))^{10} \cdot |L(q, \mathcal{T}(R))| + |\bar{B}(q, 5d_k(q, R))|)\Big),$$

where $L(\mathcal{T}(R), q)$ is the set of all performer iterations (lines 4-15) of Algorithm 4.3.

Proof. Apply Lemma 4.5 to estimate the time complexity of Algorithm F.2:
$O\big(|L(\mathcal{T}(R), q)| \cdot (\#\text{Line}[4-8] + \#\text{Line}[14-17] + \#\text{Line}[18]) + \#\text{Line}[8-14]\big)$. □

Corollary 4.7 gives a run-time bound using only minimized expansion constant $c_m(R)$, where if $R \subset \mathbb{R}^m$, then $c_m(R) \le 2^m$. Recall that $\Delta(R)$ is aspect ratio of R introduced in Definition 1.1.

Corollary 4.7. *Let R be a finite set in a metric space (X, d). Given a compressed cover tree $\mathcal{T}(R)$, Algorithm 4.3 finds all k-nearest neighbors of q in time $O\Big((c_m(R))^{10} \cdot \log_2(k) \cdot \log_2(\Delta(R)) + |\bar{B}(q, 5d_k(q, R))| \cdot \log_2(k)\Big)$.*

Proof. Replace $|L(q, \mathcal{T}(R))|$ in the time complexity of Theorem 4.6 by its upper bound from Lemma B.8: $|L(q, \mathcal{T}(R))| \le |H(\mathcal{T}(R))| \le \log_2(\Delta(R))$. □

If we are allowed to use the standard expansion constant, that corresponds to KR-dimension of Krauthgamer & Lee (2004), then we obtain a stronger result, Theorem 4.9.

Lemma F.6. *Let R be a finite reference set in a metric space (X, d) and let $q \in X$ be a query point. Let ϱ be the* special *level of $L(\mathcal{T}(R), q)$. Let $i \in L(\mathcal{T}(R), q) \setminus \varrho$ be any level. Then if $p \in R_i$ we have $d(p, q) \le 2^{i+3}$.*

Proof. By assumption in this part of the algorithm we have $d(q, \lambda) \le 2^{i+2}$. By line 7 of Algorithm F.2, since $p \in R_i$ we have $d(p, q) \le d(q, \lambda) + 2^{i+2} \le 2^{i+2} + 2^{i+2} \le 2^{i+3}$, which proves the claim. □

Lemma F.7. *Let R be a finite reference set in a metric space (X, d) and let $q \in X$ be a query point. Let ϱ be the* special *level of $L(\mathcal{T}(R), q)$. Let $i \in L(\mathcal{T}(R), q) \setminus \varrho$ be any level. Then if $p \in \mathcal{C}_i(R_{\eta(i)}) \setminus R_i$, we have $d(p, q) > 2^{i+2}$.*

Proof. By assumption $p \in \mathcal{C}_i(R_{\eta(i)}) \setminus R_i$. By line 7 of Algorithm F.2 it follows that $d(q, p) > 2^{i+2} + d(q, \lambda) \ge 2^{i+2}$. Therefore $d(q, p) > 2^{i+2}$, which proves the claim. □

Lemma F.8. *Let i be a non-minimal level of $L(\mathcal{T}(R), q)$ of Definition F.1. Assume that $t = \eta(\eta(i+3))$ is defined. Then there exists $p \in R$ satisfying $2^{i+2} < d(p, q) \le 2^{t+4}$.*

Proof. Note first that since $\eta(i+3) \in L(\mathcal{T}(R), q)$, there exists distinct $u \in R_{\eta(\eta(i+3))}$ and $v \in \mathcal{C}_{\eta(i+3)}(R_{\eta(\eta(i+3))})$, in such a way that u is the parent of v. Let us show that both of u, v cant belong to set R_i. Assume contrary that both $u, v \in R_i$. Then by Lemma F.6 we have $d(v, q) \le 2^{i+3}$ and $d(u, q) \le 2^{i+3}$. By triangle inequality $d(u, v) \le d(u, q) + d(q, v) \le 2^{i+4} \le 2^{\eta(i+3)}$. Recall that we denote a level of a node by l. On the other hand we have $l(u) \ge \eta(i+3)$ and $l(v) \ge \eta(i+3)$, by separation condition of Definition 2.1 we have $d(u, v) > 2^{\eta(i+3)}$, which is a contradiction. Therefore only one of $\{u, v\}$ can belong to R_i. It sufficies two consider the two cases below:

Assume that $v \notin R_i$. Since v is children of u we have $d(u, v) \le 2^{\eta(i+3)+1}$. By Lemma F.6 we have $d(u, q) \le 2^{\eta(\eta(i+3))+3}$. By triangle inequality

$$d(v, q) \le d(v, u) + d(u, q) \le 2^{\eta(\eta(i+3))+3} + 2^{\eta(i+3)+1} \le 2^{\eta(\eta(i+3))+4}$$

Since $v \notin R_i$ there exists level t having $\eta(i+3) \ge t \ge i$ and $v \in \mathcal{C}_t(R_{\eta(t)}) \setminus R_t$. Therefore by Lemma F.7 we have $d(q, v) > 2^{t+2} \ge 2^{i+2}$. It follows that we have found point $v \in R$ satisfying $2^{i+2} < v \le 2^{\eta(\eta(i+3))+4}$. Therefore $p = v$, is the desired point.

Assume that $u \notin R_i$. Since $u \in R_{\eta(\eta(i+3))}$, by Lemma F.6 we have $d(u, q) \le 2^{\eta(\eta(i+3))+3}$. On the other hand since $u \notin R_i$, there exists level t having $\eta(i+3) \ge t \ge i$ and $u \in \mathcal{C}_t(R_{\eta(t)}) \setminus R_t$. Therefore by Lemma F.7 we have

$d(q, u) > 2^{t+2} \geq 2^{i+2}$. It follows that we have found point $u \in R$ satisfying $2^{i+2} < u \leq 2^{\eta(\eta(i+3))+4}$. Therefore $p = u$, is the desired point.

□

Lemma 4.8. *Algorithm 4.3 executes lines 4-15 the following number of times:* $|L(\mathcal{T}(R), q)| = O(c(R \cup \{q\})^2 \cdot \log_2(|R|))$.

Proof. Let $x \in L(\mathcal{T}(R), q)$ be the lowest level of $L(\mathcal{T}(R), q)$. Define $s_1 = \eta(\eta(x)+1)$ and let $s_i = \eta(\eta(\eta(s_{i-1}+3))+3)$, if it exists. Assume that s_{n+1} is the last sequence element for which $\eta(\eta(\eta(s_{n-1}+3))+3)$ is defined. Define $S = \{s_1, ..., s_n\}$. For every $i \in \{1, ..., n\}$ let p_i be the point provided by Lemma F.8 that satisfies

$$2^{s_i+2} < d(p_i, q) \leq 2^{\eta(\eta(s_i+3))+4}.$$

Let P be the sequence of points p_i. Denote $n = |P| = |S|$. Let us show that S satisfies the conditions of Lemma 2.5. Note that:

$$4 \cdot d(p_i, q) \leq 4 \cdot 2^{\eta(\eta(s_i+3))+4} \leq 2^{\eta(\eta(s_i+3))+6} \leq 2^{\eta(\eta(\eta(s_i+3))+3)+2} \leq 2^{s_{i+1}+2} \leq d(p_{i+1}, q)$$

By Lemma 2.5 applied for $A = R \cup q$ and sequence P we get:

$$|\bar{B}(q, \frac{4}{3}d(q, p_n))| \geq (1 + \frac{1}{c(R)^2})^n \cdot |\bar{B}(q, \frac{1}{3}d(q, p_1))|$$

Since $\eta(x) \in L(\mathcal{T}(R), q)$, there exists some point $u \in R_{\eta(x)}$. By Lemma F.6 we have $d(u, q) \leq 2^{\eta(x)+3}$. Also $2^{\eta(\eta(x)+1)+1} \leq \frac{2^{\eta(\eta(x)+1)+2}}{3} < \frac{d(q,p_1)}{3}$ It follows that:

$$1 \leq |\bar{B}(q, 2^{\eta(x)+3})| \leq |\bar{B}(q, 2^{\eta(\eta(x)+1)} + 1)| \leq |\bar{B}(q, \frac{d(q, p_1)}{3})|$$

Therefore we have

$$|R| \geq \frac{|\bar{B}(q, \frac{4}{3} \cdot d(q, p_n))|}{|\bar{B}(q, \frac{1}{3} \cdot d(q, p_1))|} \geq (1 + \frac{1}{c(R \cup \{q\})^2})^n$$

Note that $c(R \cup \{q\}) \geq 2$ by definition of expansion constant. Then by applying $\log$ and by using Lemma B.7 we obtain: $c(R \cup \{q\})^2 \log(|R|) \geq n = |S|$. Let x be minimal level of $L(\mathcal{T}(R), q)$ and let y be the maximal level of $L(\mathcal{T}(R), q)$ Note that S is a sub sequence of L in such a way that:

- $[x, s_1] \cap L(\mathcal{T}(R), q) \leq 3$,
- for all $i \in 1, ..., n$ we have $[s_i, s_{i+1}] \cap L(\mathcal{T}(R), q) \leq 10$
- $[s_n, y] \cap L(\mathcal{T}(R), q) < 20$

Since segments $[x, s_1], [s_1, s_2], ..., [s_2, s_n], [s_n, y]$ cover $|L(\mathcal{T}(R), q)|$, it follows that $|S| \geq \frac{|L(\mathcal{T}(R),q)|}{20}$. We obtain that

$$|L(\mathcal{T}(R), q)| \leq 20 \cdot c(R \cup \{q\})^2 \cdot \log_2(|R|),$$

which proves the claim. □

Theorem 4.9. *Let R be a finite reference set in a metric space (X, d). Let $q \in X$ be a query point, $c(R \cup \{q\})$ be the expansion constant of $R \cup \{q\}$ and $c_m(R)$ be the minimized expansion constant from Definition 1.4. Given a compressed cover tree $\mathcal{T}(R)$, Algorithm 4.3 finds all k-nearest neighbors of q in time* $O\Big(c(R \cup \{q\})^2 \cdot \log_2(k) \cdot \big((c_m(R))^{10} \cdot \log_2(|R|) + c(R \cup \{q\}) \cdot k\big)\Big)$.

Proof. By Theorem 4.6 the required time complexity is

$$O\Big((c_m(R))^{10} \cdot \log_2(k) \cdot |L(q, \mathcal{T}(R))| + |\bar{B}(q, 5d(q, \beta))| \cdot \log_2(k)\Big)$$

for some point β among the first k-nearest neighbors of q. Apply Definition 1.4:

$$|B(q, 5d(q, \beta))| \leq (c(R \cup \{q\}))^3 \cdot |B(q, \frac{5}{8}d(q, \beta))| \tag{14}$$

Since $|B(q, \frac{5}{8}d(q, \beta))| \leq k$, we have $|B(q, 5d(q, \beta))| \leq (c(R \cup \{q\}))^3 \cdot k$. It remains to apply Lemma 4.8: $|L(q, \mathcal{T}(R))| = O(c(R \cup \{q\})^2 \cdot \log_2 |R|)$. □

Corollary F.9 combines Theorem 3.9 with Theorem 4.9, to show that Problem 1.3 can be solved in $O(c^{O(1)} \cdot \log(k) \cdot \max\{|Q|, |R|\} \cdot (\log |R|) + k)$ time.

Corollary F.9 (solution to Problem 1.3)**.** *In the notations of Theorem 4.9, set* $c = \max\limits_{q \in Q} c(R \cup \{q\})$. *Algorithms E.2 and F.2 solve Problem 1.3 in time*

$$O\Big(\max(|Q|, |R|) \cdot c^2 \cdot \log_2(k) \cdot \big((c_m(R))^{10} \cdot \log_2(|R|) + c \cdot k\big)\Big).$$

Proof. For any $q \in Q$, since $\log_2 |R \cup \{q\}| \leq 2 \log_2 |R|$, a tree $\mathcal{T}(R)$ can be built in time

$$O(c^2 \cdot c_m(R)^8 \cdot \log |R|)$$

by Theorem 3.9. Therefore the time complexity is dominated by running Algorithm F.2 on all points $q \in Q$. The final complexity is obtained by multiplying the time from Theorem 4.9 by $|Q|$. □

G. Approximate k-nearest neighbor search

The original navigating nets and cover trees were used in Krauthgamer & Lee (2004, Theorem 2.2) and Beygelzimer et al. (2006a, Section 3.2) to solve the $(1 + \epsilon)$-approximate nearest neighbor problem for $k = 1$. The main result, Theorem G.6 justifies a near linear parameterized complexity to find approximate a k-nearest neighbor set $\mathcal{P}$ formalized in Definition G.1.

Definition G.1 (approximate k-nearest neighbor set $\mathcal{P}$)**.** *Let R be a finite reference set and let Q be a finite query set of a metric space (X, d). Let $q \in Q \subseteq X$ be a query point, $k \geq 1$ be an integer and $\epsilon > 0$ be a real number. Let $\mathcal{N}_k = \cup_{i=1}^{k} \mathrm{NN}_i(q)$ be the union of neighbor sets from Definition 1.2. A set $\mathcal{P} \subseteq R$ is called an* approximate k-nearest neighbors set, *if $|\mathcal{P}| = k$ and there is an injection $f : \mathcal{P} \to \mathcal{N}_k$ satisfying $d(q, p) \leq (1 + \epsilon) \cdot d(q, f(p))$ for all $p \in \mathcal{P}$.* ■

Definition G.3 is analog of Definition F.1 for $(1 + \epsilon)$-approximate k-nearest neighbor search.

Definition G.3 (Iteration set of approximate k-nearest neighbor search)**.** *Let R be a finite subset of a metric space (X, d). Let $\mathcal{T}(R)$ be a cover tree of Definition 2.1 built on R and let $q \in X$ be an arbitrary point. Let $L(\mathcal{T}(R), q) \subseteq H(\mathcal{T}(R))$ be the set of all levels i during iterations of lines 3-19 of Algorithm G.2 launched with inputs $(\mathcal{T}(R), q)$. We denote $\eta(i) = \min_t \{t \in L(\mathcal{T}(R), q) \mid t > i\}$.* ■

Lemma G.4 (k-nearest neighbors in the candidate set for all i)**.** *Let R be a finite subset of an ambient metric space (X, d), let $q \in X$ be a query point , let $k \in \mathbb{Z} \cap [1, \infty)$ and $\epsilon \in \mathbb{R}_+$ be parameters. Let $\mathcal{T}(R)$ be a compressed cover tree of R. Assume that $|R| \geq k$. Then for any iteration $i \in L(\mathcal{T}(R), q)$ of Algorithm G.2 the candidate set $\bigcup_{p \in R_i} \mathcal{S}_i(p, \mathcal{T}(R))$ contains all k-nearest neighbors of q.* ■

Proof. Proof of this lemma is similar to Lemma G.4 and is therefore omitted. □

Lemma G.5 shows that Algorithm G.2 correctly returns an Approximate k-nearest neighbor set of Definition G.1.

Lemma G.5 (Correctness of Algorithm G.2)**.** *Algorithm G.2 finds an approximate k-nearest neighbors set of any query point $q \in X$.* ■

Proof. Assume first that condition on line 7 of Algorithm G.2 is satisfied during some iteration $i \in H(\mathcal{T}(R))$ of Algorithm G.2. Let us denote

$$\mathcal{A} = \bigcup_{p \in \mathcal{C}_i(R_{\eta(i)})} \{\mathcal{S}_i(p, \mathcal{T}(R)) \mid d(p, q) < d(q, \lambda)\}, \mathcal{B} = \bigcup_{p \in \mathcal{C}_i(R_{\eta(i)})} \{\mathcal{S}_i(p, \mathcal{T}(R)) \mid d(p, q) = d(q, \lambda)\}.$$

Algorithm G.2 This algorithm finds approximate k-nearest neighbor of Definition G.1.

1: **Input** : compressed cover tree $\mathcal{T}(R)$, a query point $q \in X$, an integer $k \in \mathbb{Z}_+$, real $\epsilon \in \mathbb{R}_+$.
2: Set $i \leftarrow l_{\max}(\mathcal{T}(R)) - 1$ and $\eta(l_{\max} - 1) = l_{\max}$. Set $R_{l_{\max}} = \{\text{root}(\mathcal{T}(R))\}$.
3: **while** $i \geq l_{\min}$ **do**
4: Assign $\mathcal{C}_i(R_{\eta(i)}) \leftarrow R_{\eta(i)} \cup \{a \in \text{Children}(p) \text{ for some } p \in R_{\eta(i)} \mid l(a) = i\}$.
5: Compute $\lambda = \lambda_k(q, \mathcal{C}_i(R_{\eta(i)}))$ from Definition D.6 by Algorithm D.8.
6: Find $R_i = \{p \in \mathcal{C}_i(R_{\eta(i)}) \mid d(q,p) \leq d(q,\lambda) + 2^{i+2}\}$.
7: **if** $\frac{2^{i+2}}{\epsilon} + 2^{i+1} \leq d(q,\lambda)$ **then**
8: Let $\mathcal{P} = \emptyset$.
9: **for** $p \in \mathcal{C}_i(R_{\eta(i)})$ **do**
10: **if** $d(p,q) < d(q,\lambda)$ **then**
11: $\mathcal{P} = \mathcal{P} \cup \mathcal{S}_i(p, \mathcal{T}(R))$
12: **end if**
13: **end for**
14: Fill $\mathcal{P}$ until it has k points by adding points from sets $\mathcal{S}_i(p, \mathcal{T}(R))$, where $d(p,q) = d(q,\lambda)$.
15: **return** $\mathcal{P}$.
16: **end if**
17: Set $j \leftarrow \max_{a \in R_i} \text{Next}(a, i, \mathcal{T}(R))$. {If such j is undefined, we set $j = l_{\min} - 1$}
18: Set $\eta(j) \leftarrow i$ and $i \leftarrow j$.
19: **end while**
20: Compute and **output** k-nearest neighbors of query point q from the set $R_{l_{\min}}$.

By Algorithm G.2 set $\mathcal{P}$ contains all points of $\mathcal{A}$ and rest of the points are filled form $\mathcal{B}$. We will now form $f : \mathcal{P} \to \mathcal{N}_k$ by mapping every point $p \in \mathcal{A} \cap \mathcal{P}$ into itself and then by extending f to be injective map on whole set $\mathcal{P}$. The claim holds trivially for all points $p \in \mathcal{A} \cap \mathcal{P}$. Let us now consider points $p \in \mathcal{P} \setminus \mathcal{A}$. Let $\gamma \in \mathcal{C}_i(R_{\eta(i)})$ be such that $p \in \mathcal{S}_i(\gamma, \mathcal{T}(R))$ and let $\psi \in \mathcal{C}_i(R_{\eta(i)})$ be such that $f(p) \in \mathcal{S}_i(\psi, \mathcal{T}(R))$. By using triangle inequality, Lemma B.6 and the fact that $p \in \mathcal{A} \cup \mathcal{B}$ we obtain:

$$d(q,p) \leq d(q,\gamma) + d(\gamma,p) \leq d(q,\lambda) + 2^{i+1} \tag{15}$$

On the other hand since $f(p) \notin \mathcal{A}$ we have

$$(1+\epsilon) \cdot d(q, f(p)) \geq (1+\epsilon) \cdot (d(q,\psi) - d(\psi, f(p))) \geq (1+\epsilon) \cdot (d(q,\lambda) - 2^{i+1}) \tag{16}$$

Note that by line 7 we have $\frac{2^{i+2}}{\epsilon} + 2^{i+1} \leq d(q,\lambda)$. It follows that $2^{i+2} \leq \epsilon \cdot d(q,\lambda) - \epsilon \cdot 2^{i+1}$. Therefore we have:

$$d(q,\lambda) + 2^{i+1} \leq d(q,\lambda) + 2^{i+2} - 2^{i+1} \leq (1+\epsilon) \cdot (d(q,\lambda) - 2^{i+1}) \tag{17}$$

By combining Equations (15) - (17) we obtain $d(q,p) \leq (1+\epsilon) \cdot d(q, f(p))$. If the condition on line 7 of Algorithm G.2 is never satisfied, then the Algorithm finds real k-nearest neighbors of point q in the end of the algorithm and therefore the claim holds.

$\square$

Theorem G.6 (Time complexity of Algorithm G.2)**.** *In the notations of Definition G.1, the complexity of Algorithm G.2 is*

$$O\Big((c_m(R))^{8+\lceil \log(2+\frac{1}{\epsilon})\rceil} \cdot \log_2(k) \cdot \log_2(\Delta(R)) + k\Big).$$

$\blacksquare$

Proof. Similarly to Lemma 4.5 it can be shown that Algorithm G.2 is bounded by:

$$O((c_m(R))^4 \cdot \log_2(k) \cdot \max_i |R_i| \cdot |H(\mathcal{T}(R))| + \#\text{Line}[7-16]) \tag{18}$$

Note first that in lines 7 - 16 we loop over set $\mathcal{C}_i(R_{\eta(i)})$ and select k points from it. Therefore $\#\text{Line}[7-16] = k + |\mathcal{C}_i(R_{\eta(i)})|$.

Let us now bound the size of R_i. By line 7 of Algorithm G.2 either Algorithm G.2 is launched that terminates the program or $\frac{2^{i+2}}{\epsilon} + 2^{i+1} > d(q,\lambda)$. Let C_i be the ith cover set of $\mathcal{T}(R)$. To bound $|R_i|$ we can assume the latter. Similarly to Theorem 4.9 we have:

$$R_i = \{r \in \mathcal{C}_i(R_{\eta(i)}) \mid d(p,q) \leq d(q,\lambda) + 2^{i+2}\} \tag{19}$$

$$= \bar{B}(q, d(q,\lambda) + 2^{i+2}) \cap \mathcal{C}_i(R_{\eta(i)}) \tag{20}$$

$$\subseteq \bar{B}(q, d(q,\lambda) + 2^{i+2}) \cap C_i \tag{21}$$

$$\subseteq \bar{B}(q, 2^{i+2}(\frac{3}{2} + \frac{1}{\epsilon})) \cap C_i \tag{22}$$

Since the cover set C_i is a 2^i-sparse subset of the ambient metric space X, we can apply Lemma 2.2 with $t = 2^{i+2}(\frac{3}{2} + \frac{1}{\epsilon})$ and $\delta = 2^i$. Since $4\frac{t}{\delta} + 1 = 2^4(\frac{3}{2} + \frac{1}{\epsilon}) + 1 \leq 2^4(2 + \frac{1}{\epsilon})$, we get $\max|R_i| \leq (c_m(R))^{4+\lceil \log_2(2+\frac{1}{\epsilon})\rceil}$. The final complexity is obtained by plugging the upper bound of $|R_i|$ above into (18). □

Corollary G.7 (complexity for approximate k-nearest neighbors set $\mathcal{P}$)**.** *In the notations of Definition G.1, an approximate k-nearest neighbors set is found for all $q \in Q$ in time $O\Big(|Q| \cdot (c_m(R))^{8+\lceil \log(2+\frac{1}{\epsilon})\rceil} \cdot \log(k) \cdot \log_2(\Delta(R)) + |Q| \cdot k\Big)$.* ■

Proof. This corollary follows directly from Theorem G.6 . □

H. Discussions: current contributions and future steps

This paper rigorously proved the time complexity of the exact k-nearest neighbor search. The motivations were the past gaps in the proofs of time complexities in Beygelzimer et al. (2006a, Theorem 5), Ram et al. (2009, Theorem 3.1), March et al. (2010, Theorem 5.1). Though Elkin & Kurlin (2022a) provided concrete counterexamples, no corrections were published. Main Theorem 4.9 and Corollary 3.10 have finally filled the above gaps.

To overcome all past obstacles, first Definition 1.2 and Problem 1.3 rigorously dealt with a potential ambiguity of k-nearest neighbors at equal distances, which was not discussed in the past work.

A new compressed cover tree in Definition 2.1 substantially simplified the navigating nets Krauthgamer & Lee (2004) and original cover trees Beygelzimer et al. (2006a) by avoiding any repetitions of given data points. This compression has substantially clarified the construction and search Algorithms E.2 and F.2.

Second, section C showed that the new minimized expansion constant c_m of any finite subset R of a normed vector space $\mathbb{R}^n$ has the upper bound 2^m. In the future, it can be similarly shown that if R is uniformly distributed then classical expansion constant $c(R)$ is 2^m as well.

Third, sections E and F corrected the approach of Beygelzimer et al. (2006a) as follows. Assuming that expansion constants and aspect ratio of a reference set R are fixed, Corollaries 3.10 and F.9 rigorously showed that the times are linear in the maximum size of R, Q and near-linear $O(k \log k)$ in the number k of neighbors.

The future problem is to improve the complexity of k-nearest neighbor search to a pure linear time $O(c(R)^{O(1)}|R|$ by using cover trees on both sets Q, R. Since a similar approach Ram et al. (2009) was shown to have incorrect proof in Elkin & Kurlin (2022a, Counterexample 6.5) and Curtin et al. (2015); Elkin & Kurlin (2022b) used additional parameters I, θ, this goal will require significantly more effort to understand if $O(c(R)^{O(1)}|R|)$ is achievable by using a compressed cover tree.

Corollary F.9 allowed us to justify the near-linear time of generically complete PDD Widdowson & Kurlin (2021) invariants (Pointwise Distance Distributions), which recently distinguished all (more than 660 thousand) periodic crystals in the world's largest database of real materials Widdowson et al. (2022). Due to these ultra-fast invariants, more than 200 billion pairwise comparisons were completed over two days on a modest desktop while past tools were estimated to require over 34 thousand years Widdowson & Kurlin (2022). The huge speed of PDD is complemented by slower but provably complete invariant isosets Anosova & Kurlin (2021) with continuous metrics that allow polynomial-time approximations Anosova & Kurlin (2022).