Labeled Nearest Neighbor Search and Metric Spanners via Locality Sensitive Orderings

Arnold Filtser*[1]

[1]Bar-Ilan University

Abstract

Chan, Har-Peled, and Jones [SICOMP 2020] developed locality-sensitive orderings (LSO) for Euclidean space. A (τ,ρ)-LSO is a collection Σ of orderings such that for every $x, y \in \mathbb{R}^d$ there is an ordering $\sigma \in \Sigma$, where all the points between x and y w.r.t. σ are in the ρ-neighborhood of either x or y. In essence, LSO allow one to reduce problems to the 1-dimensional line. Later, Filtser and Le [STOC 2022] developed LSO's for doubling metrics, general metric spaces, and minor free graphs.

For Euclidean and doubling spaces, the number of orderings in the LSO is exponential in the dimension, which made them mainly useful for the low dimensional regime. In this paper, we develop new LSO's for Euclidean, ℓ_p, and doubling spaces that allow us to trade larger stretch for a much smaller number of orderings. We then use our new LSO's (as well as the previous ones) to construct path reporting low hop spanners, fault tolerant spanners, reliable spanners, and light spanners for different metric spaces.

While many nearest neighbor search (NNS) data structures were constructed for metric spaces with implicit distance representations (where the distance between two metric points can be computed using their names, e.g. Euclidean space), for other spaces almost nothing is known. In this paper we initiate the study of the labeled NNS problem, where one is allowed to artificially assign labels (short names) to metric points. We use LSO's to construct efficient labeled NNS data structures in this model.

*Email: `arnold.filtser@biu.ac.il`. This research was supported by the ISRAEL SCIENCE FOUNDATION (grant No. 1042/22).

Contents

1 Introduction

1.1 Locality Sensitive Ordering

Chan, Har-Peled, and Jones [CHJ20] recently introduce a new and powerful tool into the algorithmist's toolkit, called *locality sensitive ordering* (abbreviated LSO). LSO provides an order over the points of a metric space (X, d_X), this order being very useful, as it helps to store, sort, and search the data (among other manipulations).

Definition 1 ((τ,ρ)-LSO)**.** *Given a metric space (X, d_X), we say that a collection Σ of orderings is a (τ,ρ)-LSO if $|\Sigma| \le \tau$, and for every $x, y \in X$, there is a linear ordering $\sigma \in \Sigma$ such that (w.l.o.g.[1]) $x \preceq_\sigma y$ and the points between x and y w.r.t. σ could be partitioned into two consecutive intervals I_x, I_y where $I_x \subseteq B_X(x, \rho \cdot d_X(x,y))$ and $I_y \subseteq B_X(y, \rho \cdot d_X(x,y))$. ρ is called the* stretch *parameter.*

Morally speaking, given a problem, LSO can reduce it from a general and complicated space to a much simpler space: 1-dimensional line. Chan *et al.* [CHJ20] constructed $\left(O_d(\epsilon^{-d}) \cdot \log\frac{1}{\epsilon}, \epsilon\right)$-LSO for the d-dimensional Euclidean space. They used their LSO to design simple dynamic algorithms for approximate nearest neighbor search, bichromatic closest pair, MST, spanners, and fault-tolerant spanners. Later, Buchin, Har-Peled, and Oláh [BHO19, BHO20] constructed reliable spanners using LSO, obtaining considerably superior results compared with previous techniques.

Filtser and Le [FL22] generalized Chan *et al.* [CHJ20] result to doubling spaces,[2] showing that every metric space with doubling dimension d admits a $\left(\epsilon^{-O(d)}, \epsilon\right)$-LSO. Furthermore, they generalized the concept of LSO to other metric spaces, defining the two related notions of *triangle-LSO* (which turn to be useful for general metric spaces), and *left-sided LSO* (which turn to be useful for topologically restricted graphs). Here, instead of presenting the left-sided LSO's of [FL22], we introduce the closely related notion of *rooted-LSO*, which has some additional structure. All the results and constructions for left-sided LSO in [FL22] hold for rooted LSO as well. We refer to [FL22] for a comparison between the different notions, and to Figure 1 for an illustration.

Definition 2 ((τ,ρ)-Triangle-LSO)**.** *Given a metric space (X, d_X), we say that a collection Σ of orderings is a (τ,ρ)-triangle-LSO if $|\Sigma| \le \tau$, and for every $x, y \in X$, there is an ordering $\sigma \in \Sigma$ such that (w.l.o.g.[1]) $x \prec_\sigma y$, and for every $a, b \in X$ such that $x \preceq_\sigma a \preceq_\sigma b \preceq_\sigma y$ it holds that $d_X(a,b) \le \rho \cdot d_X(x,y)$.*

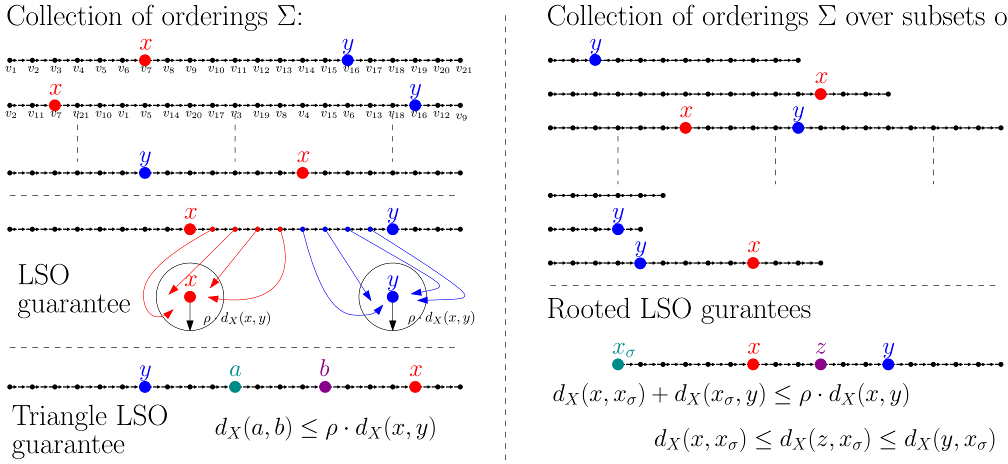
Definition 3 ((τ,ρ)-rooted-LSO)**.** *Given a metric space (X, d_X), we say that a collection Σ of* orderings *over subsets of X is a (τ,ρ)-rooted-LSO if the following hold:*

- *Each point $x \in X$ belongs to at most τ orderings in Σ.*
- *Each ordering $\sigma \in \Sigma$ is associated with a point $x_\sigma \in X$, which is the first in the order, and such that the ordering is w.r.t. distances from x_σ (i.e. $y \prec_\sigma z \Rightarrow d_X(x_\sigma, y) \le d_X(x_\sigma, z)$).*
- *For every pair of points u, v, there is some $\sigma \in \Sigma$ containing both x, y, and such that $d_G(u, x_\sigma) + d_G(x_\sigma, v) \le \rho \cdot d_G(u,v)$.*

Filtser and Le [FL22] constructed triangle LSO for general metrics, and rooted LSO for the shortest path metrics of trees, treewidth graphs, planar graphs, and graph excluding a fixed minor. They used their LSO's to construct oblivious reliable spanners for the respective metric spaces, considerably improving previous constructions (that used different techniques). All the known results on LSO's are summarized in Table 1.

[1] That is either $x \preceq_\sigma y$ or $y \preceq_\sigma x$, and the guarantee holds w.r.t. all the points between x and y in the order σ.

[2] A metric space (X, d) has doubling dimension d if every ball of radius $2r$ can be covered by 2^d balls of radius r.



Collection of orderings Σ:
x
y
v_1 v_2 v_3 v_4 v_5 v_6 v_7 v_8 v_9 v_{10} v_{11} v_{12} v_{13} v_{14} v_{15} v_{16} v_{17} v_{18} v_{19} v_{20} v_{21}
x
y
v_2 v_{11} v_7 v_{21} v_{10} v_1 v_5 v_{14} v_{20} v_{17} v_3 v_{19} v_8 v_4 v_{15} v_6 v_{13} v_{18} v_{16} v_{12} v_9
y
x
x
y
LSO guarantee
x
$\rho \cdot d_X(x,y)$
y
$\rho \cdot d_X(x,y)$
y
a
b
x
Triangle LSO guarantee
$d_X(a,b) \leq \rho \cdot d_X(x,y)$
Collection of orderings Σ over subsets of X
y
x
x
y
y
y
x
Rooted LSO gurantees
x_σ
x
z
y
$d_X(x,x_\sigma) + d_X(x_\sigma,y) \leq \rho \cdot d_X(x,y)$
$d_X(x,x_\sigma) \leq d_X(z,x_\sigma) \leq d_X(y,x_\sigma)$

Figure 1: Illustration of different types of LSO.

Previously constructed LSO for the Euclidean space [CHJ20], as well as for metric spaces with doubling dimension d [FL22], have exponential dependency on the dimension in their cardinality, a phenomena often referred to as "the curse of dimensionality". When the dimension is high, it can be a major obstacle. Indeed, the distances induced by n point in an $O(\log n)$-dimensional Euclidean space induce a metric space which is much more structured than a general metric space. Therefore one might expect them to admit better LSO. However, using [CHJ20] one can only construct (n,ϵ)-LSO (note that every metric admits $(\lceil\frac{n}{2}\rceil,0)$-LSO [3]).

Every n point metric space has doubling dimension at most $\log n$. Consider the case where the doubling dimension is somewhat large (e.g. $\sqrt{\log n}$) but not maximal. It is much more structured than general metric, however the only construction we have [FL22] gives us $\epsilon^{-O(d)}$ orderings, which might be too large. If we insist that the number of orderings will be small, could we take advantage of the doubling structure to construct better LSO then for general metrics?

Our contribution. In this paper we construct new triangle-LSO for high dimensional spaces. We then present many applications for the newly constructed LSO's, as well as for the previously constructed LSO's. Previous and new LSO construction are summarized in Table 1.

Theorem 1. *For every $t \in [4, 2\sqrt{d}]$, $\delta \in (0,1]$, and $d \geq 1$, the d-dimensional Euclidean space $\mathbb{R}^d$ admits $\left(O\left(\frac{d^{1.5}}{\delta \cdot t} \cdot \log(\frac{2\sqrt{d}}{t}) \cdot \log\frac{d}{\delta} \cdot e^{\frac{d}{2t^2}\cdot(1+\frac{2}{t^2})}\right), (1+\delta)t\right)$-triangle LSO.*

For $t = \frac{2}{3}\sqrt{d}$ and $\delta = \frac{1}{2}$, we obtain $\left(O(d\log d), \sqrt{d}\right)$-triangle LSO. In particular, for every set of n points in ℓ_2, using the Johnson Lindenstrauss dimension reduction [JL84], for every fixed $t > 1$, we can construct $\left(n^{\frac{1}{t^2}} \cdot \tilde{O}(\frac{\log^{1.5} n}{t}), O(t)\right)$-triangle LSO, or $\left(\tilde{O}(\log n), O(\sqrt{\log n})\right)$-triangle LSO, a quadratic improvement compared with general n-point metric spaces!

Interestingly, in Corollary 1 we show that the $\left(O(d\log d), \sqrt{d}\right)$-triangle LSO Σ for ℓ_2, is in the same time also a $\left(O(d\log d), d^{\frac{1}{p}}\right)$-triangle LSO for ℓ_p where $p \in [1,2]$, and $\left(O(d\log d), d^{1-\frac{1}{p}}\right)$-triangle LSO for ℓ_p where $p \in [2,\infty]$. For $p \in [1,2]$, we generalize Theorem 1 to ℓ_p spaces to get the

[3] This follows from a theorem by Walecki [Als08] who showed that the edges of the K_n clique graph can be partitioned into $\lceil\frac{n}{2}\rceil$ Hamiltonian paths.

LSO type	Metric Space	# of orderings (τ)	Stretch (ρ)	Ref
(Classic) LSO	Euclidean space $\mathbb{R}^d$	$O_d(\epsilon^{-d})\cdot\log\frac{1}{\epsilon}$ $^{(*)}$	ϵ	[CHJ20]
	Doubling dimension d	$\epsilon^{-O(d)}$	ϵ	[FL22]
Triangle-LSO	General metric	$O(n^{\frac{1}{k}}\cdot\log n\cdot\frac{k^2}{\epsilon}\cdot\log\frac{k}{\epsilon})$	$2k+\epsilon$	[FL22]
	Euclidean space $\mathbb{R}^d$	$e^{\frac{d}{2t^2}\cdot(1+\frac{2}{t^2})}\cdot\tilde{O}(\frac{d^{1.5}}{\epsilon\cdot t})$	$(1+\epsilon)t$	Thm. 1
	ℓ_p^d for $p\in[1,2]$	$e^{O(d/t^p)}\cdot\tilde{O}(d)$	t	Thm. 2
	ℓ_p^d for $p\in[2,\infty]$	$\tilde{O}(d)$	$d^{1-\frac{1}{p}}$	Cor. 1
	Doubling dimension d	$2^{O(d/t)}\cdot d\cdot\log^2 t$	t	Thm. 3
Rooted LSO	Tree	$\log n$	1	[FL22]
	Treewidth k	$k\cdot\log n$	1	[FL22]
	Planar / fixed minor free	$O(\frac{1}{\epsilon}\cdot\log^2 n)$	$1+\epsilon$	[FL22]

Table 1: Summary of all known results, on all the different types of locality sensitive orderings (LSO). $k\in\mathbb{N}$, $t>1$, $\epsilon\in(0,1)$ is an arbitrarily small parameter. $^{(*)}$ O_d hides an arbitrary function of d, the number of orderings in [CHJ20] LSO is $O_d(\epsilon^{-d})\cdot\log\frac{1}{\epsilon}=2^{O(d)}\cdot d^{\frac{3}{2}d}\cdot\epsilon^{-d}\cdot\log\frac{1}{\epsilon}$.

entire #ordering-stretch trade-off. Finally, we generalize Theorem 1 to general metric spaces with doubling dimension d.

Theorem 2. *For every $p\in[1,2]$, $t\in[5,d^{\frac{1}{p}}]$ and $d\geq 1$, the d-dimensional ℓ_p space admits $\left(e^{O(\frac{d}{t^p})}\cdot\tilde{O}(d),t\right)$-triangle LSO.*

Theorem 3. *Given a metric space (X,d_X) with doubling dimension d, and parameter $t\in[\Omega(1),d]$, X admits $\left(2^{O(d/t)}\cdot d\cdot\log^2 t,t\right)$-triangle LSO.*

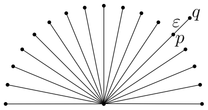
For $t=d$, we get $(\tilde{O}(d),d)$-triangle LSO, again much better then general metric spaces!

1.2 Labeled Nearest Neighbor Search

Nearest neighbor search (abbreviated NNS) is a classical and fundamental task used in numerous domains including machine learning, clustering, document retrieval, databases, statistics, data compression, database queries, computational biology, data mining, pattern recognition, and many others. In the NNS problem we are given a set P of points in a metric space (X,d_X). The goal is to construct a succinct data structure that given a query point $q\in X$, quickly returns a point $p\in P$ closest to q (i.e. arg $\min_{p\in P} d_X(p,q)$). In order to keep the size of the data structure, and the query time small, usually approximation is allowed. In the t-approximate nearest neighbor problem (abbreviated t-NNS) the goal is to return a point p at distance at most $t\cdot\min_{p\in P} d_X(p,q)$ from q. The problem was extensively studied in ℓ_p spaces (see the survey [AIR18]), and also in various norm spaces over $\mathbb{R}^d$ (see e.g. [ANN$^+$17, ANRW21]). NNS data structures were also constructed beyond normed spaces. Some examples are Earth-Mover distance [IT03], Edit Distance [OR07, AIR18], and Fréchet distance [Ind02, DS17, EP18, FFK20]. We observe that a crucial property shared by these examples, is that they have an "implicit distance representation". That is, it is possible to compute the distance between two points using only their names (e.g. the coordinates values in $\mathbb{R}^d$ used as names: $d_{\mathbb{R}^d}((x_1,\dots,x_d),(y_1,\dots,y_d))=\|(x_1,\dots,x_d)-(y_1,\dots,y_d)\|_2$).

For general metric spaces, Krauthgamer and Lee [KL05] introduced the *black box model*. Here one is given access to an exact distance oracle [4] DO that answer distance queries in t_{DO} time. They showed that one can construct an efficient $1+\epsilon$-NNS (that is with polynomial space, and polylogarithmic query time), if and only if the doubling dimension of X is at most $O(\log\log n)$.

Indeed, for metric spaces with large doubling dimension, distance queries provide very limited information. Consider for example the case where the input metric is the star graph (inducing uniform metric on the leaves, see illustration on the right), and the query point attached to one of the leaves with an edge of infinitesimal weight, one must query all the points before finding any finite approximation to the nearest neighbor.



An interesting case studied by Abraham, Chechik, Krauthgamer, and Wieder [ACKW15] is that of planar graphs. Here we are given a huge weighted planar graph $G=(V,E,w)$ with N vertices, and a subset of n vertices $X\subseteq V$. The goal is to solve the $(1+\epsilon)$-NNS problem w.r.t. the shortest path metric d_G, input set X and queries from V. Assuming access to an exact distance oracle[4] DO that answer distance queries in t_{DO} time, and given a planar graph G of maximum degree Δ, Abraham *et al.* [ACKW15] constructed a $(1+\epsilon)$-NNS data structure for planar graph of size $n\cdot O(\epsilon^{-1}\cdot\log\log N+\Delta\cdot\log^2 n)$ and query time $O((\epsilon^{-1}\cdot\log\log n+t_{\mathrm{DO}})\cdot\log\log N+\log n\cdot\Delta\cdot t_{\mathrm{DO}})$.

Linear dependence on the degree is a very limiting requirement, as planar graphs have apriori unbounded degree. Moreover, exact distance computations (even in planar graphs) are time consuming, and if the graph is big enough could be infeasible. Exact distance oracle is a highly non-trivial assumption, it is an expensive data structure, [5] better to be avoided. One might hope to relax either the max degree assumption, or to use the much more reasonable and efficient data structure of approximate distance oracle [Tho04, Kle02, LW21]. Unfortunately, Abraham *et al.* [ACKW15] showed both assumptions to be necessary. Specifically, the dependence on the degree is necessary, as every NNS data structure with space at most $O(\frac{N}{\Delta\log_\Delta n})$ must probe the distance oracle at least $\Omega(\Delta\log_\Delta n)$ times. Furthermore, they show that if one is only given access to a $(1+\epsilon)$-distance oracle, then there is a planar graph (in fact a tree) with maximum degree $O(\log n)$, aspect ratio $O(\frac{\log n}{\epsilon})$, $N\le n^2$, and the NNS data structure is forced to make $\Omega(n)$ queries to the distance oracle.

To conclude this discussion, exact distance oracle (assumed both by the black box model [KL05] and [ACKW15]) is an expensive data structure, which enables us to construct efficient NNS only under very limiting assumptions (small doubling dimension / constant maximum degree in planar graphs). From the other hand in many metric spaces with "implicit distance representation" efficient NNS were constructed. The crux is that the information stored in the name (e.g. coordinate values) used to preform various manipulations on the data, in addition to distance computation. What if in planar graphs, or even in completely general metric spaces, we could choose the names of the metric points, or alternatively assigning each point a short label, would it be possible to construct efficient NNS data structures?

To answer this question, we introduce the *labeled t-NNS* problem.

Definition 4 (Labeled t-NNS). *Consider an N-point metric space (X,d_X), where one can assign to each point $x\in X$ an arbitrary short label l_x. Given a subset $P\subseteq X$ of size n (unknown in advance)*

[4] An exact distance oracle D is a data structure that given two points x,y, returns $\mathrm{est}(x,y)=d_X(x,y)$. A distance oracle of stretch t returns a value $\mathrm{est}(x,y)$ in $[d_X(x,y),t\cdot d_X(x,y)]$.

[5] After a long line of work, the state of the art (by Long and Pettie [LP21]) requires either super-linear space $N^{1+o(1)}$, or very large query time $N^{o(1)}$, both quite undesirable.

together with their labels $\{l_x\}_{x\in P}$ (but without access to (X,d_X) or any additional information) the goal is to construct a NNS search data structure as follows: given a query $q\in X$ together with its assigned label ℓ_q, the data structure will return a t-approximate nearest neighbor $p\in P$: $d_X(p,q)\le t\cdot\min_{x\in P}d_X(x,q)$. The parameters of study are: label size, data structure size, query time, and approximation factor t.

We also consider the scenario where the set P is changing dynamically: points are added and removed from P. Here we are required to maintain a data structure for P, while minimizing the update time (as well as all the other parameters).[6]

In the labeled NNS model we get to assign a short label (alternatively choose a name) for each point in a big metric space (X,d_X). These labels tries to imitate the natural hint provided by the name of the points themselves in metric spaces with implicit distance representation. The main object of study here is the trade-off between label size, and the approximation of the resulting NNS. A trivial choice of label for each point x will be simply to store distances to all other points. However the label size $\Omega(N)$ is infeasible. A more sophisticated solution is the following: fix constants $k,t\in\mathbb{N}$, and embed all the points in (X,d_X) into $d=\tilde{O}(N^{\frac{1}{k}})$-dimensional ℓ_∞ [Mat96, ABN11]. That is we assign each point x a vector $v_x\in\mathbb{R}^d$ such that $\forall x,y\in X$, $d_X(x,y)\le\|v_x-v_y\|_\infty\le(2k-1)\cdot d_X(x,y)$, and use the vectors as labels. Given an n point subset $P\subseteq X$ with its respective labels (vectors), use Indyk's NNS [Ind01] over $\{v_x\}_{x\in P}$ to construct a NNS data structure $\mathbb{D}_{\text{Ind}}$ with approximation factor $O(\log_{1+\frac{1}{t}}\log d)=O(t\cdot\log\log N)$ w.r.t. the ℓ_∞ vectors, space $\tilde{O}(d\cdot n^{1+\frac{1}{t}})=\tilde{O}(N^{\frac{1}{k}}\cdot n^{1+\frac{1}{t}})$ and query time $\tilde{O}(n^{1+\frac{1}{k}})$. Given a query q, we will simply query $\mathbb{D}_{\text{Ind}}$ on the vector v_q, and on answer v_p will return p. Note that the query time and space are the same as above, while the approximation factor will be $O(k\cdot t\cdot\log\log N)$.

Our contribution. Our results for the labeled t-NNS are summarized in Table 2. We begin by proving meta Theorem 7 showing that (τ,ρ)-rooted LSO implies a labeled ρ-NNS with label size $O(\tau)$, space $O(n\cdot\tau)$, query time $O(\tau)$, and update time $O(\tau\cdot\log\log N)$. As a result we conclude efficient labeled $(1+\epsilon)$-NNS data structures for fixed minor free graphs (and planar), and exact labeled NNS for treewidth graphs. Another interesting corollary is an efficient labeled NNS for metrics with small *correlation dimension* (a generalization of doubling, see [CG12]).

Next, we prove a meta Theorem 8, showing that (τ,ρ)-triangle LSO implies a labeled 2ρ-NNS with label size $O(\tau\cdot\log N)$, space $O(n\cdot\tau\cdot\log N)$, and query and update time $O(\tau\cdot\log\log N)$. We conclude an efficient labeled NNS for graphs with large doubling dimension. For the high-dimensional Euclidean space, approximate nearest neighbor search was extensively studied (see the survey [AIR18], and additional discussion in Section 1.5). However, for the case of doubling metrics, NNS never went beyond $1+\epsilon$ approximation. In particular, in all existing solutions the query time and space have exponential dependence on the dimension (see Section 1.5). Thus ours are the first results in this regime, removing "the curse of dimensionality".

As an additional corollary of the triangle LSO to labeled NNS meta theorem one can derive a NNS of for general metric spaces which considerably improved upon the labeled NNS based on [Mat96]+[Ind01] discussed above. However, the query time turns out to be somewhat large. We provide direct constructions for labeled NNS for general metrics, getting label size $\tilde{O}(\epsilon^{-1}\cdot N^{\frac{1}{k}})$, stretch $8(1+\epsilon)k$ and very small query time: $O(\epsilon^{-1}\cdot\log\log N)$ (Theorem 12). We show that the

[6] For example, suppose that we have a NNS data structure for a set P. Dynamic NNS, should be able to efficiently update the data structure to work w.r.t. a slightly updated set $P'=P\cup\{x\}\setminus\{y\}$ instead of P.

Family	stretch	label	query time	update time	Ref
Planar / Minor free	$1+\epsilon$	$O(\epsilon^{-1}\log^2 N)$	$O(\epsilon^{-1}\log^2 N)$	$\epsilon^{-1}\cdot\tilde{O}(\log^2 N)$	Cor. 2
Treewidth k	1	$O(k\log N)$	$O(k\log N)$	$k\cdot\tilde{O}(\log N)$	Cor. 3
Correlation dim. k	$1+\epsilon$	$\tilde{O}_{k,\epsilon}(\sqrt{N})$	$\tilde{O}_{k,\epsilon}(\sqrt{N})$	$\tilde{O}_{k,\epsilon}(\sqrt{N})$	Cor. 4
Ultrametric	1	$O(\log N)$	$O(\log\log N)$	$O(\log\log N)$	Thm. 4
General Metric	$8(1+\epsilon)k$	$O(\frac{k}{\epsilon}N^{\frac{1}{k}}\cdot\log N)$	$O(\frac{1}{\epsilon}\cdot\log\log N)$	$O(\frac{k}{\epsilon}N^{\frac{1}{k}}\cdot\log\log N)$	Thm. 12
	$t<2k+1$	$\tilde{\Omega}(N^{\frac{1}{k}})$	arbitrary	arbitrary	Thm. 15
Doubling dim. d	t	$2^{O(d/t)}\cdot\tilde{O}(d)\cdot\log N$	$2^{O(d/t)}\cdot\tilde{O}(d)\cdot\log\log N$	$2^{O(d/t)}\cdot\tilde{O}(d)\cdot\log\log N$	Cor. 5

Table 2: Labeled NNS data structures for different families. The second to last line (Theorem 15) is a lower bound. Space is measured in machine words. The label size and query time always equal. The space in all the cases above equals n times the label size.

standard information theoretic bound applies for the labeled NNS as well, specifically, for stretch $t<2k+1$, the label size must be $\tilde{\Omega}(n^{\frac{1}{k}})$ (regardless of query time, Theorem 15). Finally, we put special focus on the regime where the stretch is $O(\log N)$. We obtain labeled NNS scheme with very short label and small query time. Most notably, assuming polynomial aspect ratio, and allowing the bound on the label to be only in expectation, we can obtain $O(1)$ label size, and $O(\log\log N)$ query time. See Table 7 for a summary.

1.3 Spanners

Given a metric space (X,d_X), a metric *spanner* is a graph H over X points, such that that the shortest path metric d_H in H, closely resembles the metric d_X. Formally, a *t-spanner* for X is a weighted graph $H(X,E,w)$ that has $w(u,v)=d_X(u,v)$ for every edge $(u,v)\in E$ and $d_H(x,y)\leq t\cdot d_X(x,y)$ for every pair of points $x,y\in X$. [7] The classic parameter of study is the trade-off between stretch and sparsity (number of edges). Althöfer *et al.* [ADD+93] showed that every n point metric space admits a $2k-1$ spanner with $O(n^{1+\frac{1}{k}})$ edges, while every set of n points in $\mathbb{R}^d$, or more generally metric space of doubling dimension d, admits a $(1+\epsilon)$-spanner with $n\cdot\epsilon^{-O(d)}$ edges [DHN93, GGN06]. We refer to the book [NS07], and the survey [ABS+20] for an overview.

Path Reporting Low Hop Spanners Recently, Kahalon, Le, Milenkovic, and Solomon [KLMS22] studied *path reporting low-hop spanners*. While a t-spanner guarantees that a "short" path exists between every two points, such a path might be very long, and finding it is a time consuming operation. A path reporting t-spanner, is a spanner accompanied with a data structure that given a query pair $\{x,y\}$, efficiently retrieves a path between x and y (of total weight $\leq t\cdot d_X(x,y)$). A path P with h edges is called an h-hop path. H is an h-hop t-spanner of X if for every $x,y\in X$, there is an h-hop path P from x to y in H, such that $w(P)\leq t\cdot d_X(x,y)$. Clearly, the time required to report a path is at least as large as the number of edges along the path, thus we wish to minimize the number of hops.

Low number of hops is a highly desirable property in network design, as each transmission causes delays, which are non-negligible when the number of transmissions is large [AT11, BF18].

[7] Frequently the literature is concerned with graph spanners, where given a graph $G=(V,E,w)$ the goal is to find a subgraph H preserving distances. Here we study metric spanners, where there is no underlying graph.

Metric family	stretch	sparsity	query time	Ref
General Metric	$O(k)$	$O\left(n^{1+\frac{1}{k}}\cdot k\cdot\log n\right)$	$O(1)$	[KLMS22]
	$2k-1$	$O(n^{1+\frac{1}{k}}\cdot k)$	$O(k)$	Thm. 21, [TZ05]
	$(1+\epsilon)(4k-2)$	$O(n^{1+\frac{1}{k}}\cdot\epsilon^{-1}\cdot k\cdot\log\Phi)$	$O(\epsilon^{-1}\cdot\log 2k)$	Thm. 22
Doubling Dimension d	$1+\epsilon$	$\epsilon^{-O(d)}\cdot n\cdot\log n$	$\epsilon^{-O(d)}$	[KLMS22]
	t	$2^{-O(d/t)}\cdot\tilde{O}(n)$	$2^{-O(d/t)}\cdot d\cdot\log^2 t$	Cor. 11
Euclidean $\mathbb{R}^d$	$1+\epsilon$	$O_d(\epsilon^{-d})\cdot\log\frac{1}{\epsilon}\cdot n\cdot\log n$	$O_d(1)$	Cor. 12
	$(1+\epsilon)t$	$\tilde{O}(\frac{d^{1.5}}{\epsilon\cdot t})\cdot e^{\frac{2d}{t^2}\cdot(1+\frac{8}{t^2})}\cdot n\log n$	$\tilde{O}(\frac{d^{1.5}}{\epsilon\cdot t})\cdot e^{\frac{2d}{t^2}\cdot(1+\frac{8}{t^2})}$	Cor. 11
ℓ_p^d, $p\in[1,2]$	t	$\tilde{O}(d)\cdot e^{O(\frac{d}{t^p})}\cdot n\log n$	$\tilde{O}(d)\cdot e^{O(\frac{d}{t^p})}$	Cor. 11
ℓ_p^d, $p\in[2,\infty]$	$2\cdot d^{1-\frac{1}{p}}$	$\tilde{O}(d)\cdot n\log n$	$\tilde{O}(d)$	Cor. 11
Tree	1	$O\left(n\cdot\log n\right)$	$O(1)$	[KLMS22]
Fixed Minor Free	$1+\epsilon$	$O\left(n\cdot\epsilon^{-2}\cdot\log^3 n\right)$	$O(\epsilon^{-2}\cdot\log^2 n)$	[KLMS22]
	$1+\epsilon$	$O(n\cdot\epsilon^{-1}\cdot\log^2 n)$	$O(\epsilon^{-1}\cdot\log n)$	Cor. 13
Planar	$1+\epsilon$	$O(n\cdot\epsilon^{-1}\cdot\log^2 n)$	$O(\epsilon^{-1})$	Cor. 14
Treewidth k	1	$O(n\cdot k\cdot\log n)$	$O(k)$	Cor. 15

Table 3: Summary of old and new results on path reporting low hop spanners. The spanners are for n point metrics, and all report paths with hop bound 2. Here $\epsilon\in(0,1)$, $k,d\geq 1$ are integers. The space required for the path reporting data structure is asymptotically equal to the sparsity of the spanner in all the cases other than Euclidean space Corollary 12, where there is an additional additive factor of $O_d(\epsilon^{-2d})\log\frac{1}{\epsilon}$.

Low hop networks are also known to be more reliable [BF18, WA88, RAJ12], and used in electricity and telecommunications [BF18], and many other (practical) network design problems [LCM99, BA92, GPdSV03, GM03, PS03]. Hop-constrained network approximation is often used in parallel computing [Coh00, ASZ20], as the number of hops governs the number of required parallel rounds (e.g. in Dijkstra).

Kahalon *et al.* [KLMS22] constructed path reporting low-hop spanners for many spaces, such as path reporting 2-hop $O(k)$-spanners with $O(n^{1+\frac{1}{k}}\cdot k\cdot\log n)$ edges, and $O(1)$ query time for general metrics, and path reporting 2-hop $(1+\epsilon)$-spanners with $O(\frac{n}{\epsilon^2}\cdot\log^3 n)$ edges and $O(\epsilon^{-2}\cdot\log^2 n)$ query time for planar graphs. They showed a plethora of applications for their spanners: compact routing schemes, fault tolerant routing, spanner sparsification, approximate shortest path trees (SPT), minimum weight trees (MST), and online MST verification.

Our contribution. Kahalon *et al.* [KLMS22] first constructed path reporting low hop spanners for trees, and then reduced each type of metric to the case of trees. We observe that it is actually enough to reduce to the even simpler case of paths, and obtain a host of such spanners using LSO's. We then manually improve some of the resulting spanners, most notably we create path reporting 2-hop $(1+\epsilon)$-spanner for planar graph with $O(\frac{n}{\epsilon}\log^2 n)$ edges and $O(\frac{1}{\epsilon})$-query time, and a path reporting 2-hop $(1+\epsilon)$-spanner for points in d-dimensional Euclidean space with $O_d(\epsilon^{-d})\cdot\log\frac{1}{\epsilon}\cdot n\log n$ edges and $O_d(1)$-query time. See Table 3 for a summary of old and new results.

Family	Stretch	Edges	Ref
Doubling dimension d	$1+\epsilon$	$\epsilon^{-O(d)} \cdot f^2 \cdot n \cdot \log n$	[KLMS22]
	$1+\epsilon$	$\epsilon^{-O(d)} \cdot f \cdot n \cdot \log n$	Cor. 16
	t	$2^{-O(d/t)} \cdot f \cdot \tilde{O}(n)$	Cor. 16
General Metric	$4k+\epsilon$	$\tilde{O}(n^{1+\frac{1}{k}} \cdot f \cdot \epsilon^{-1})$	Cor. 16
Euclidean $\mathbb{R}^d$	$1+\epsilon$	$O_d(\epsilon^{-d}) \log \frac{1}{\epsilon} \cdot f \cdot n \cdot \log n$	Cor. 16
	$(1+\epsilon)k$	$e^{\frac{2d}{k^2} \cdot (1+\frac{8}{k^2})} \cdot \tilde{O}(\frac{d^{1.5}}{\epsilon \cdot k}) \cdot f \cdot n \cdot \log n$	Cor. 16
ℓ_p^d, $p \in [1,2]$	k	$e^{O(\frac{d}{k^p})} \cdot \tilde{O}(d) \cdot f \cdot n \cdot \log n$	Cor. 16
ℓ_p^d, $p \in [2,\infty]$	$2 \cdot d^{1-\frac{1}{p}}$	$\tilde{O}(d) \cdot f \cdot n \cdot \log n$	Cor. 16
Treewidth k	2	$O(n \cdot k \cdot f \cdot \log n)$	Cor. 16
Fixed Minor Free	$2+\epsilon$	$O(\frac{n}{\epsilon} \cdot f \cdot \log^2 n)$	Cor. 16

Table 4: Summary of old and new results on 2-hop f-fault tolerant spanners. The spanners are for n point metrics, and all report paths with hop bound 2. Here $\epsilon \in (0,1)$, $k, d \geq 1$ are integers.

Fault tolerant spanners Levcopoulos, Narasimhan, and Smid [LNS02] introduced the notion of a fault-tolerant spanner. A graph $H = (X, E_H, w)$ is an f-*vertex-fault-tolerant* t-spanner of a metric space (X, d_X), if for every set $F \subset X$ of at most f vertices, it holds that $\forall u, v \notin F$, $d_{H \setminus F}(u,v) \leq t \cdot d_X(u,v)$. For general metrics, after a long line of work [CLPR10, DK11, BDPW18, BP19, DR20, BDR21, Par22], it was shown that every n-vertex graph admits an efficiently constructible f-vertex-fault-tolerant $(2k-1)$-spanner with $O(f^{1-1/k} \cdot n^{1+1/k})$ edges, which is optimal assuming the Erdös' Girth Conjecture [Erd64]. For n-points in d dimensional Euclidean space, or more generally in a space of doubling dimension d, f-*vertex fault tolerant* $(1+\epsilon)$-spanner were constructed with $\epsilon^{-O(d)} \cdot f \cdot n$ edges [LNS02, Luk99, Sol14].

Kahalon *et al.* [KLMS22] initiated the study of low-hop fault tolerant spanners (previous constructions had $\Omega(\log n)$ hops). An h-hop f-fault tolerant t-spanner H of a metric (X, d_x) is a graph over X such that for every set $F \subseteq X$ of at most f vertices, for every $x, y \notin F$, the spanner without F: $H[X \setminus F]$ contains an h-hop path between x to y of weight at most $t \cdot d_X(x,y)$. The advantages of such a spanner are straightforward, we refer to [KLMS22] for a discussion. Kahalon *et al.* constructed a 2-hop f-fault tolerant spanner for doubling spaces with $n \cdot f^2 \cdot \epsilon^{-O(d)} \cdot \log n$ edges. Note that a linear dependence on f is necessary (as if a point has degree $\leq f$ in H, we can delete all it's neighbors and get distortion ∞). It is natural to ask whether it is possible to construct such a spanner with only a linear dependence, and not quadratic as in [KLMS22].

Our contribution. One can easily construct f-fault tolerant 1-spanner for the path graph with $O(nf)$ edges. We observe that using $O(nf \log n)$ edges, it is possible to obtain f-fault tolerant 2-hop 1-spanner for the path graph (note that $O(n \log n)$ edges are necessary for every 2-hop spanner [AS87, LMS22]). Using the various old and new LSO's, we obtain a host of f-fault tolerant 2-hop spanners for various metric spaces. Most notably, for metrics with doubling dimension d, we obtain an f-fault tolerant 2-hop $(1+\epsilon)$-spanner with $\epsilon^{-O(d)} \cdot f \cdot n \cdot \log n$ edges, getting the desired linear dependence on f. See Table 4 for a summary of results.

Family	stretch	guarantee	size	ref
Euclidean $(\mathbb{R}^d, \|\cdot\|_2)$	$1+\epsilon$	Deterministic	$n \cdot \tilde{O}_d(\epsilon^{-7d})\nu^{-6} \cdot \tilde{O}(\log n)$	[BHO19]
	$1+\epsilon$	Oblivious	$n \cdot \tilde{O}_d(\epsilon^{-2d}) \cdot \tilde{O}(\nu^{-1}(\log\log n)^2)$	[BHO20]
	$(1+\epsilon)t$	Oblivious	$\nu^{-1} \cdot e^{\frac{4d}{t^2}\cdot(1+\frac{8}{t^2})} \cdot \tilde{O}(n \cdot \frac{d^3}{\epsilon^2 \cdot t^2})$	Cor. 17
ℓ_p^d for $p \in [1,2]$	t	Oblivious	$\nu^{-1} \cdot e^{O(\frac{d}{t^p})} \cdot \tilde{O}(n \cdot d^2)$	Cor. 17
ℓ_p^d for $p \in [2,\infty]$	$2 \cdot d^{1-\frac{1}{p}}$	Oblivious	$\nu^{-1} \cdot \tilde{O}\left(n \cdot d^2\right)$	Cor. 17
Doubling dimension d	$1+\epsilon$	Deterministic	$n \cdot \epsilon^{-O(d)}\nu^{-6} \cdot \tilde{O}(\log n)$	[FL22]
	$1+\epsilon$	Oblivious	$n \cdot \epsilon^{-O(d)}\nu^{-1}\log\nu^{-1} \cdot \tilde{O}(\log\log n)^2$	[FL22]
	t	Oblivious	$\tilde{O}(n) \cdot \nu^{-1} \cdot 2^{O(d/t)}$	Cor. 17
General metric	$8t+\epsilon$	Oblivious	$\tilde{O}(n^{1+1/t} \cdot \epsilon^{-2}) \cdot \nu^{-1}$	[FL22]
Tree	2	Oblivious	$n \cdot O(\nu^{-1}\log^3 n)$	[FL22]
Treewidth k	2	Oblivious	$n \cdot O(\nu^{-1}k^2\log^3 n)$	[FL22]
Planar/Minor-free	$2+\epsilon$	Oblivious	$n \cdot O(\nu^{-1}\epsilon^{-2}\log^5 n)$	[FL22]

Table 5: Summary of previous and new constructions of ν-reliable spanners.

Reliable spanners A major limitation of fault tolerant spanners is that the number of failures must be determined in advance. In particular, such spanners cannot withstand a massive failure. One can imagine a scenario where a significant portion (even 90%) of a network fails and ceases to function (due to, e.g., close-down during a pandemic), it is important that the remaining parts of the network (or at least most of it) will remain highly connected and functioning. To this end, Bose *et al.* [BDMS13] introduced the notion of a *reliable spanner*. A ν-reliable spanner is a graph such that for every failure set $B \subseteq X$, the residual spanner $H \smallsetminus B$ is a t-spanner for $X \smallsetminus B^+$, where $B^+ \supseteq B$ is a superset of cardinality at most $(1+\nu) \cdot |B|$. An oblivious ν-reliable t-spanner is a distribution $\mathcal{D}$ over spanners, such that for every failure set B, $H \smallsetminus B$ is a t-spanner for $X \smallsetminus B_H^+$, where the superset B_H^+ depends on both B and the sampled spanner H. The guarantee is that the cardinality of B_H^+ is bounded by $(1+\nu) \cdot |B|$ in expectation, see Definition 10.

ν-Reliable spanners were constructed for d dimensional Euclidean and doubling spaces with $n \cdot \epsilon^{-O(d)} \cdot \tilde{O}(\log n)$ edges [BHO19, BHO20, FL22] by a reduction from (classic) LSO's. Oblivious reliable spanners were constructed also for planar, minor free, treewidth graphs, and general metrics [FL22] by reductions from triangle, and rooted LSO's (as well as from sparse covers [HMO21]).

Our contribution. Our newly constructed triangle LSO's for high dimensional Euclidean, ℓ_p spaces, and doubling spaces, directly imply reliable spanners for these spaces, obtaining the first results without exponential dependence on the dimension. See Table 5 for a summary.

Light spanners An extensively studied parameter is the *lightness* of a spanner, defined as the ratio $w(H)/w(MST(X))$, where $w(H)$ resp. $w(MST(X))$ is the total weight of edges in H resp. a minimum spanning tree (MST) of X. Obtaining spanners with small lightness (and thus total weight) is motivated by applications where edge weights denote e.g. establishing cost. The best possible total weight that can be achieved in order to ensure finite stretch is the weight of an MST, thus making the definition of lightness very natural.

Obtaining light spanners for general graphs has been the subject of an active line of work

	Stretch	**Lightness**	
Euclidean space	$O(t)$	$O(n^{\frac{1}{t^2}}\cdot\log n\cdot t)$	[LS21]
	$O(t)$	$O(e^{\frac{d}{t^2}}\cdot\log^2 n\cdot t)$	[FN22]
	$(1+\epsilon)2t$	$e^{\frac{d}{2t^2}\cdot(1+\frac{2}{t^2})}\cdot\tilde{O}(\frac{d^{1.5}}{\epsilon^2})\cdot\log n$	Cor. 19
	$(1+\epsilon)4t$	$e^{\frac{d}{2t^2}\cdot(1+\frac{2}{t^2})}\cdot\tilde{O}(\frac{d^{1.5}}{\epsilon^2})\cdot\log^* n$	Cor. 19
Doubling dimension d	$O(t)$	$O(2^{\frac{d}{t}}\cdot t\cdot\log^2 n)$	[FN22]
	$O(t)$	$2^{O(d/t)}\cdot d\cdot\log^2 t\cdot\log^* n$	Cor. 18
	d	$O(d\cdot\log^2 n)$	[FN22]
	d	$O(d\cdot\log^2 d\cdot\log^* n)$	Cor. 18
ℓ_p^d for $p\in[1,2]$	t	$O(\frac{t^{1+p}}{\log^2 t}\cdot n^{O(\frac{\log^2 t}{t^p})}\cdot\log n)$	[FN22]
	t	$e^{O(\frac{d}{t^p})}\cdot\tilde{O}(d\cdot t)\cdot\log^* n$	Cor. 20
ℓ_p^d for $p\in[2,\infty]$	$4\cdot d^{1-\frac{1}{p}}$	$\tilde{O}(d^{2-\frac{1}{p}})\cdot\log^* n$	Cor. 21

Table 6: Summary of previews and new results of light spanners for high dimensional metric spaces. Interestingly, for $p\in[1,2]$ [FL22] obtain lightness $O(\frac{t^{1+p}}{\log^2 t}\cdot n^{O(\frac{\log^2 t}{t^p})}\cdot\log n)$ regardless of dimension, which is superior to our Corollary 20 for $d\gg\log n$.

[ADD+93, CDNS95, ENS15, BFN19b, CW18, FS20], where the state of the are by Le and Solomon [LS21] who obtained $(1+\epsilon)(2k-1)$ spanner with lightness $O(\epsilon^{-1}\cdot n^{\frac{1}{k}})$. Light spanners were also studied extensively in Euclidean spaces (see the book [NS07]), doubling spaces [Got15, FS20, BLW19], planar and minor free graphs [Kle06, Kle08, BLW17, Le20, CFKL20], and high dimensional Euclidean and doubling spaces [HIS13, FN22, LS21].

Our contribution. Recently Le and Solomon [LS21] obtain a general framework for constructing light spanners from spanner oracles. We construct new spanner oracles using LSO's. As a result we derive new light spanners, that improve the state of the art for high dimensional spaces (and match the state of the art for low dimensional doubling spaces). See Table 6 for a summary of results.

1.4 Technical ideas

Triangle LSO for high dimensional Euclidean space Our construction is very natural: partition the space randomly in every distance scale ξ_i (for some large ξ) into clusters of diameter ξ_i, such that close-by points are likely to be clustered together. In the created ordering σ, points in each cluster will be ordered consecutively and recursively. In particular, the ordering σ will correspond to a laminar partition obtained by the clustering in all possible scales. For a pair of points $x,y\in\mathbb{R}^d$ to be satisfied in the resulting ordering σ, they have to be clustered together in all the distance scales $\xi^i\ge t\cdot\|x-y\|_2$.

Our space partition in each scale is done using ball carving (ala [AI08]): pick a uniformly random series of centers $z_1,z_2,\dots,$. Each points is assigned to the cluster of the first center at distance at most $R=\frac{1}{2}\cdot\xi^i$. We show that a finite random seed of size $d^{O(d)}$ is enough to sample such a clustering (in all possible distance scales, simultaneously). The probability that two points x,y are clustered together is than equal to the ratio between the volumes of intersection and union of balls: $\Pr[x,y \text{ clustered together}] = \frac{\mathrm{Vol}_d(B(x,R)\cap B(y,R))}{\mathrm{Vol}_d(B(x,R)\cup B(y,R))} \geq \Omega(\frac{1}{\sqrt{d}})\cdot\left(1-(\frac{\|x-y\|_2}{R})^2\right)^{d/2}$. We bound this ratio for the case $\|x-y\|_2 \leq \frac{R}{\sqrt{d}}$ using a lemma from [CCG+98] (see Fact 1). For the general case, we prove that the ratio between these volumes is at least $\Omega(\frac{R}{\sqrt{d}\cdot\|p-q\|_2})\cdot(1-(\frac{\|p-q\|_2}{R})^2)^{\frac{d}{2}}$ (Fact 2), slightly improving a similar fact from [And09], by a $\frac{R}{\|p-q\|_2}$ factor. This ratio eventually governs our success probability (when replacing $R/\|p-q\|_2$ by twice the stretch $2t$). The improved analysis of the volumes ratio is significant for the $O(\sqrt{d})$-stretch regime, improving the number of orderings to $\tilde{O}(d)$ (compered with $\tilde{O}(d^{1.5})$ orderings if we were using [And09]).

To generalize this construction to ℓ_p spaces, we use the exact same construction, replacing ℓ_2 balls with ℓ_p balls. The volume ratio lemma from [CCGG98] for close-by points is replaced by a crude observation (Observation 1) without any significant consequences to the resulting number of orderings. For the general case, we directly analyze the ratio of volumes for ℓ_p-balls (our computation is similar to [Ngu13]). The rest of the analysis is the same.

Triangle LSO for doubling spaces Ultrametrics (Definition 6) are trees with additional structure, where each ultrametric admits a $(1,1)$-triangle LSO. (τ,ρ)-ultrametric cover (Definition 7) of a metric space (X,d_X) is a collection $\mathcal{U}$ of τ ultrametrics such that every pair $x,y\in X$ is well approximated by the ultrametrics: $d_X(x,y)\leq\min_{U\in\mathcal{U}}d_U(x,y)\leq\rho\cdot d_X(x,y)$. Filtser and Le [FL22] showed that (τ,ρ)-ultrametric cover implies (τ,ρ)-triangle LSO. We construct $\left(2^{O(d/t)}\cdot d\cdot\log^2 t, t\right)$-ultrametric cover for spaces with doubling dimension d, which implies Theorem 3.

Our starting point for constructing the ultrametric cover is Filtser's [Fil19] padded partition cover (Definition 5), which is a collection of $\approx 2^{O(d/t)}$ space partitions where all clusters are of diameter at most Δ, and every ball of radius $\frac{\Delta}{t}$ is fully contained in a single cluster in one of the partitions. We take a single partition from each distance scales, where the gap between the distance scales is somewhat large: $O(\frac{t}{\epsilon})$. Initially these partitions are unrelated, and we "force" them to be laminar (see Figure 3), while keeping the padding property. Each such laminar partition induces an ultrametric, and their union is the desired ultrametric cover.

Labeled NNS Morally, given a (τ,ρ) LSO (or triangle LSO), the NNS label of every point is simply its position in each ordering. Given a query q, we simply find its successor and predecessor in each one of the orderings, one of them is guaranteed to be an approximate nearest neighbor (abbreviated ANN). We can find the successor and predecessor in each ordering in $O(\log\log N)$ time using Y-fast trie [Wil83], it only remains to choose one of the 2τ candidates to be the ANN. To solve this problem we again deploy the LSO structure, and construct a 2-hop 1-spanner for the implicit path graph induced by each ordering. Specifically, each point will be associated with $O(\log N)$ edges (the name and weight of which will be added to the NNS label), where given two points $x\prec_\sigma y$, in $O(1)$ time we will be able to find a point z such that $x\preceq_\sigma z\preceq_\sigma y$ and x and y stored $\{x,z\},\{y,z\}$ respectively (see Theorem 4). Then $d_X(x,z)+d_X(z,y)$ will provide us the desired estimate of $d_X(x,y)$, which will be used to choose the ANN.

The case of rooted LSO is simpler- the label of each point z will consist of its position in all the orderings σ it belongs to, and the distance to the first point x_σ (w.r.t. d_X). Given a query q, for each ordering σ containing q, the leftmost point $y_\sigma\in P$ in the ordering will be a candidate ANN. We will estimate the distance from q to y_σ by $d_X(q,x_\sigma)+d_X(x_\sigma,y_\sigma)$, and return the point with minimum estimation.

For general metrics, the number of orderings is polynomial, $N^{\frac{1}{k}}$ which results in similar NNS label size, and query time (following the approach above). While the NNS label essentially cannot be improved (Theorem 15), we can significantly reduce the query time. Our solution is to use Ramsey trees [BLMN05b, MN07, Bar21, ACE+20], which are a collection of embeddings into ultrametrics $\mathcal{U}$ such that each point x has a single home ultrametric $U_x \in \mathcal{U}$ which well approximate all the distances to x. We thus reduce the labeled NNS problem to ultrametrics, where it can be efficiently solved. For the case of approximation factor $O(\log N)$ the required number of ultrametrics is $O(\log N)$, which leads us to label size $O(\log^2 N)$. To reduce it even farther, we use the novel clan embedding [FL21], where instead of embedding the space X into a collection of ultrametrics, we embed it into a single ultrametric (but where each point might have several copies). This allows us to reduce the label size to $O(\log N)$ (in expectation), and with one additional easing assumption (either polynomial aspect ratio or small failure probability) to even $O(1)$ label size, see Table 7.

Path reporting low hop spanners A (τ,ρ)-tree cover is similar to ultrametric cover discussed above, where the ultrametrics are replaced by trees. Kahalon *et al.* [KLMS22] first constructed path reporting low hop spanner for a tree metric, and then for each metric space of interest, they considered it's tree cover, and constructed a path reporting low hop spanner for each tree in the cover. The spanner for the global metric is obtained by taking the union of all these spanners constructed for the trees in the cover. To report a queried distance, they simply computed the paths in all the trees, and returned the shortest observed path.

Thus Kahalon *et al.* idea is to reduce the problem to the fairly simple case of tree metrics. We reduce each metric space into the even simpler case of paths using LSO. Given an LSO (or triangle LSO) we simply construct a path reporting 2-hop path for each path associated with an ordering, and similarly to [KLMS22], check all the path spanners and return the shortest observed path. The resulting query time has linear dependence on the number of orderings. The case of rooted LSO is simpler, where it is enough to add a single edge per ordering, to the leftmost point in the ordering.

Next we present some improvements to the query. First, for the case of Euclidean space (low dimensional), we observe that given two points x,y, the ordering satisfying them could be computed in $O_d(1)$ time, implying that we don't need to check all the orderings, and return a 2 hop path in $O_d(1)$ time. Next, for the case of planar graphs, using the structure of cycle separators (which are used to construct the rooted LSO), in $O(1)$ time one can narrow the number of potential orderings to $O(\epsilon^{-1})$, implying $O(\epsilon^{-1})$ query time. For general graphs we observe that the celebrated Thorup Zwick distance oracle [TZ05] can be used to produce a path reporting 2-hop $(2k-1)$-spanner with $O(n^{1+\frac{1}{k}}\cdot k)$ edges and $O(k)$ query time. Finally, we use sparse covers [AP90] to obtain an exponential improvement in the query time, while incurring a factor 2 increase in the stretch.

Fault tolerant spanners The 2-hop f-fault tolerant spanner for doubling metrics by Kahalon *et al.* [KLMS22] is based on a quite sophisticated tool of robust tree cover. We have a superior, and an extremely simple construction. First we observe that the path graph has a 2-hop f-fault tolerant 1-spanner with $O(nf\log n)$ edges. Indeed, add edges from all the vertices to the middle $f+1$ vertices, delete the middle vertices and recurse on each side. We then apply this construction on each of the path graphs induced by the LSO (or triangle LSO) to obtain our results. The case of rooted LSO is even simpler: for every path it is enough to add all the edges to the first $f+1$ points.

1.5 Related Work

Most of the literature on NNS studies the (r,c)-Near Neighbor problem, where given a query q, the goal is to find a neighbor p at distance at most $t\cdot r$ from q, or declare that there are no points at distance at most r from q. Har-Peled, Indyk, and Motwani [HIM12] gave a general reduction from the (r,c)-Near Neighbor problem to $(1+\epsilon)\cdot c$-NNS with a blow up of $O(\log n)$ in the query time, $O(\epsilon^{-1}\cdot\log n)$ in space, and $O(\log n)$ in the failure probability. In this paper we do not go though this reduction, and solve the actual problem directly. In particular, for general metric spaces we can get query time much better that the additional $O(\log n)$ inflicted by the reduction.

Nearest neighbor search for doubling dimension d spaces is well studied [KL04, HM06, CG06, HK13]. One can find a $(1+\epsilon)$-approximated nearest neighbor with space $2^{O(d)}\cdot n$, and query time $2^{O(d)}\log n+\epsilon^{-O(d)}$ both being exponentially dependent on the dimension. Note that one can retrieve similar performance in the labeled NNS model using the approach in this paper. No previous work attempt to solve the problem for moderately large doubling dimension.

For the high-dimensional Euclidean space, approximate nearest neighbor search was extensively studied (see the survey [AIR18]), where the state of the art is by Andoni and Razenshteyn [AR15] who solve the r-near neighbor problem with approximation c, space $n^{1+\rho}+O(nd)$ and query time $n^{\rho}+dn^{o(1)}$, where $\rho=\frac{1}{2c^2-1}$. Using the triangle-LSO's presented in this paper, we can obtain a similar result (though with worse constants). Specifically, first use the JL transform [JL84] to reduce the dimension to $O(\log n)$, then use the (hard-coded) $(\tilde{O}(e^{\frac{d}{c^2}}),c)\approx(\tilde{O}(n^{\frac{1}{c^2}}),O(c))$-triangle LSO, and proceed as in Theorem 8 (given a query, check it's neighbors in all orderings and take the closest one). The result will be an $O(c)$ approximation with $\tilde{O}(n^{1+\frac{1}{c^2}})$ space and $\tilde{O}(n^{\frac{1}{c^2}})$ query time.

Sparse (and light) spanners were constructed efficiently in different computational models such as standard RAM [ES16, EN19a, ADF+22, LS22], LOCAL [KPX08], CONGEST [EFN20], streaming [Elk11, Bas08, BS07, KW14, AGM12, FKN21], massive parallel computation (MPC) [BDG+21] and dynamic graph algorithms [BKS12, BFH19], and online [BFKT22]. Path reporting distance oracles were studied by Elkin and Pettie [EP15] (see also [ENS15]), who constructed a distance oracle of space $O(n^{1+\frac{1}{k}})$, and that given a query pair, in $O(n^{\epsilon})$ time returns a path with stretch $O_{\epsilon}(k)$.

An extensively studied related notion to low hop metric spanners is *hop-set*. Given a graph $G=(V,E,w)$, a hop-set is a set of edges H' that when added to G, every pair of vertices have a low-hop path. Formally, an (t,h)-hop-set is a set H' of edges such that $\forall u,v\in V$, $d_G(u,v)\le d_{G\cup H'}(u,v)$, and $G\cup H'$ contains an h-hop path P between u,v of weight $w(P)\le t\cdot d_G(u,v)$. Elkin and Neiman [EN19b] constructed $\left(1+\epsilon,\left(\frac{\log k}{\epsilon}\right)^{\log k-2}\right)$ hop-sets with $O_{\epsilon,k}(n^{1+\frac{1}{k}})$ edges. We refer to the survey [EN20] for further details. Another related notion is low-hop emulator [ASZ20], where the h-hop distance is required to respect the triangle inequality.

2 Preliminaries

All logarithms (unless explicitly stated otherwise) are in base 2. We use $\tilde{O}$ notation to hide polylogarithmic factors. That is $\tilde{O}(s)=O(s)\cdot\log^{O(1)}(s)$. Similarly, we use O_d notation to hide arbitrary factors in d. That is $O_d(s)=O(s)\cdot f(d)$ for some arbitrary function $f:\mathbb{N}\to\mathbb{N}$.

In the data structures constructed in this paper, the space is counted in machine words and not bits (unless explicitly stated otherwise). Machine word is generally thought of as being $O(\log N)$ bits, and can store number, identifier, pointer etc. In addition, we consider storing a given label as taking $O(1)$ time (as it is just storing a pointer to the place where it is actually stored).

Let (X,d_X) be a metric space. The aspect ratio (sometimes called spread) $\Phi = \frac{\max_{x,y\in X} d_X(x,y)}{\min_{x\neq y\in X} d_X(x,y)}$ is the ratio between maximum and minimum distances. This paper studies metric spaces, however we also study graphs. Note that a metric space can be viewed as a complete undirected graph with edge weights respecting the triangle inequality. Given a weighted graph $G=(V,E,w)$ where the weight function $w:E\to\mathbb{R}_{\geq 0}$ gets positive values, the shortest path metric d_G is a metric space where the distance between two vertices u,v equals to the minimal weight of a path from u to v. Frequently the literature is concerned with graph spanners, where given a graph $G=(V,E,w)$ the goal is to find a subgraph H preserving distances. Here we study metric spanners, where there is no underlying graph. For example, when we discuss spanners for planar graphs (or other topologically restricted graph families), we assume that the underlying metric is the shortest path metric of a planar graph G, but we are allowed to take any edge, regardless of whether it is in G. Interesting to note, for the case of the lightness parameter, if the metric comes from a shortest path metric of a graph G, given a metric spanner H for d_G, we can always turn H into a graph spanner H' (a subgraph of G) of the same lightness as H (this is by replacing each edge e in H by the shortest path between its endpoints in G). Thus our result in Section 6.4 hold for graph spanners as well.

A metric (X,d_X) has doubling dimension d if every ball of radius r can be covered by at most 2^d balls of radius $r/2$. Doubling dimension is a generalization of the concept of Euclidean dimension to general metric spaces. In particular, the d-dimensional Euclidean space has doubling dimension $\Theta(d)$. A key property of doubling metrics (see, e.g., [GKL03]) is the *packing property* that grantees that for a subset of points $S\subseteq X$ with minimum interpoint distance r that is contained in a ball of radius R, it holds that $|S|=\left(\frac{2R}{r}\right)^{O(d)}$.

The path graph P_n contains n vertices $v_1,v_2,\dots,v_n$ and there is (unweighted) edge between v_i and v_j iff $|i-j|=1$. A path $v_{i_1},v_{i_2},\dots v_{i_s}$ is monotone iff for every j, $i_j<i_{j+1}$. Note that if a spanner H contains a monotone path between v_i,v_j then $d_H(v_i,v_j)=d_{P_n}(v_i,v_j)=|i-j|$. The following result is usually attributed to Alon and Schieber [AS87] (see also [BTS94, Sol13]), we provide a proof as we will use properties not always explicitly stated.

Theorem 4 ([AS87])**.** *The path graph $P_n=(v_1,v_2,\dots,v_n)$ admits a 2-hop 1-spanner H with $O(n\cdot\log n)$ edges. Moreover, there is a path reporting data structure with constant query time. Specifically, each vertex v_i is responsible for a subset of edges E_i incident on v_i where $|E_i|=O(\log n)$. Given a pair $i<j$, in $O(1)$ time, we can find an $l\in[i,j]$ such that $\{v_i,v_l\}\in E_i$ and $\{v_j,v_l\}\in E_j$.*

Proof. For simplicity we will assume that $n=2^\delta$ for some integer δ. Later we can simply construct the spanner for a power of 2 in $[n,2n)$. Assume by induction that for every $\delta'<\delta$, a 2-hop 1-spanner with $n'\log n'+1=2^{\delta'}\cdot\delta'+1$ edges and the data structure above, exist for the path graph $P_{n'}=P_{2^{\delta'}}$. The base case with 2 vertices and $\delta'=0$ clearly holds. Next consider $n=2^\delta$. Add edges from all the vertices to $v_{2^{\delta-1}}$ (and for every vertex v_i add $\{v_i,v_{2^{\delta-1}}\}$ to E_i). Next use the induction hypothesis on the graph induced by $(v_1,v_2,\dots,v_{2^{\delta-1}})$, and the graph induced by $(v_{2^{\delta-1}+1},\dots,v_{2^\delta})$. By the induction hypothesis, the total number of edges is bounded by

$$2^\delta-1+2\cdot\left(2^{\delta-1}\cdot(\delta-1)+1\right)=2^\delta\cdot\delta+1=n\cdot\log n+1\ .$$

As the depth of the recursion is $\log n$, clearly for every i, $|E_i|\leq\log n$. To report a 2 hop path between v_a and v_b, we simply need to find the maximal i such that $a\leq c\cdot 2^i\leq b-1$ for some integer

c. The path will be $(v_a, v_{c\cdot 2^i}, v_b)$, where $\{v_a, v_{c\cdot 2^i}\} \in E_a$ and $\{v_b, v_{c\cdot 2^i}\} \in E_b$. To find the number $c \cdot 2^i$, we can simply look on the binary representation of $a = \alpha_1\alpha_2 \dots \alpha_s$ and $b - 1 = \beta_1\beta_2 \dots \beta_s$, and let j be the most significant bit where they differ, then $c \cdot 2^i = \beta_1 \dots \beta_j 0 \dots 0$. This operation take $O(1)$ time in the standard RAM computational model. □

3 LSO's for high dimensional Euclidean space

This section is devoted to proving Theorem 1, restated bellow for convenience:

Theorem 1. *For every $t \in [4, 2\sqrt{d}]$, $\delta \in (0,1]$, and $d \geq 1$, the d-dimensional Euclidean space $\mathbb{R}^d$ admits $\left(O\left(\frac{d^{1.5}}{\delta\cdot t} \cdot \log(\frac{2\sqrt{d}}{t}) \cdot \log\frac{d}{\delta} \cdot e^{\frac{d}{2t^2}\cdot(1+\frac{2}{t^2})}\right), (1+\delta)t\right)$-triangle LSO.*

In Sections 3.1 to 3.3 we describe the construction of our LSO, and prove it's properties. In Section 3.4 we generalize the construction for ℓ_p spaces for $p \in [1,2]$.

3.1 The construction of the LSO

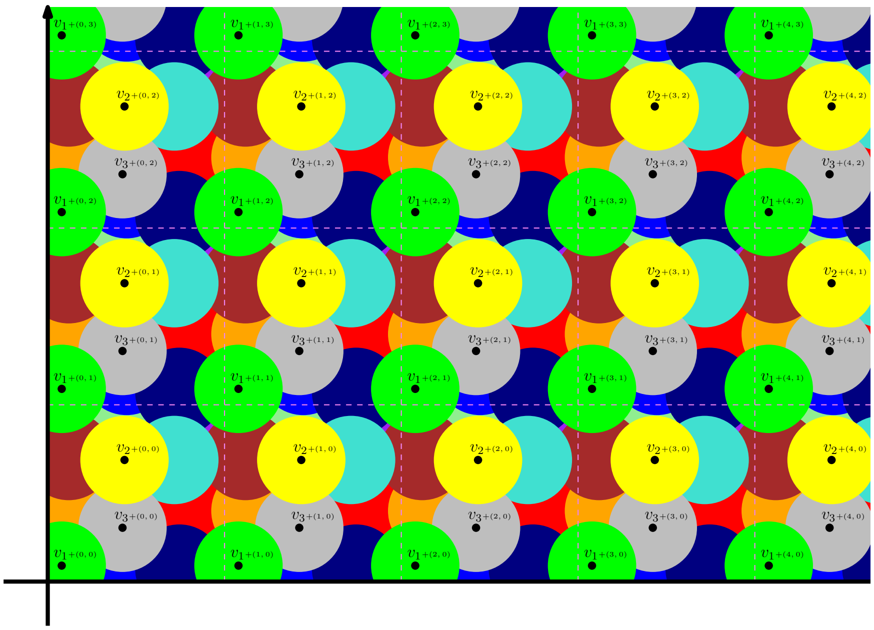
Fix a stretch parameter $t \geq 1$. Given a distance scale $w > 0$ we define the following random clustering procedure (introduced by Andoni and Indyk [AI08]), where in addition there will be an order over the clusters. The point set $4w \cdot \mathbb{Z}^d = \left\{(4w\cdot\alpha_1, \dots, 4w\cdot\alpha_d) \in \mathbb{R}^d \mid \alpha_1, \dots, \alpha_d \in \mathbb{Z}\right\}$ is the d-dimensional axis parallel lattice with side length $4w$. $G^{d,w} = \cup_{x\in 4w\cdot\mathbb{Z}^d} B(x,w)$ is the union of all the balls with radius w and center at a lattice point $4w \cdot \mathbb{Z}^d$. Given a point $v \in \mathbb{R}^d$, $v + G^{d,w}$ is simply $G^{d,w}$ shifted by v, we will denote this set by $G_v^{d,w}$. We create clusters $\mathcal{C}$ of $\mathbb{R}^d$ as follows: let $v_1, v_2, \dots, v_l \in [0, 4\cdot w)^d$ random points chosen independently (i.i.d) and uniformly, until $\bigcup_{i=1}^l G_{v_i}^{d,w} = \mathbb{R}^d$. By a standard volume argument (see also Lemma 3.2.2. in [And09]), $l = 2^{O(d\log d)}$ with high probability. For every v_i, and $u \in \mathbb{Z}^d$ there will be a cluster $C_{v_i,u}$ containing all the points p in the ball $B(v_i + 4w\cdot u, w)$ with center $v_i + 4w\cdot u$ and radius w, such that there is no $v_{i'}$ with $i' < i$ and $u' \in \mathbb{Z}^d$ such that $p \in B(v_{i'} + 4w\cdot u', w)$. The set of created clusters is denoted $\mathcal{C}_w$. See Figure 2 for illustration. We define an order $\preceq_{\mathcal{C}^w}$ over the clusters where $C_{v_i,u} \preceq_{\mathcal{C}^w} C_{v_{i'},u'}$ if either $i < i'$, or $i = i'$ and $u \preceq_{\text{lex}} u'$ in the lexicographical order (defined on vectors in $\mathbb{Z}^d$).

Next, we define a random order $\preceq$ over the entire space $\mathbb{R}^d$. For every $i \in \mathbb{Z}$, let $w_i = \xi^i$ for $\xi = \frac{12\sqrt{d}}{t}$ (which will denote the gap between two consecutive scales), and let $\mathcal{C}^{w_i}$ be a clustering drawn randomly as above. Given two different points p, q, note that they must belong to different clusters for small enough i. Let $i_{p,q}$ be the maximal index such that p and q belong to different clusters $C_p, C_q \in \mathcal{C}^{w_{i_{p,q}}}$. Then $p \preceq q$ iff $C_p \preceq_{\mathcal{C}^{w_{i_{p,q}}}} C_q$, and $q \preceq p$ otherwise. [8] The random order $\preceq$ is now defined.

Computational aspects Our LSO is defined w.r.t the entire space $\mathbb{R}^d$. Formally, given a pair of points, one is required to check infinite number of scales to find the first place they are separated. In all the applications in this paper we will be dealing with finite n-point subsets. Here it will be enough to create $\log\Phi$ scales of clusters (Φ is the aspect ratio), and there is no need to define the LSO w.r.t. the entire $\mathbb{R}^d$. Thus computing the order between a pair of points could be done efficiently (though exponential in d time).

[8] If the clusterings $\mathcal{C}^{w_i}$ are drawn independently for each scale i, such a maximal index $i_{p,q}$ exist with probability 1. However, we will construct clusters for different scales with dependencies, and insure that every pair will have such an index.

Figure 2: *Illustration of the clustering process for $w = \frac{1}{4}$ and $d = 2$. First the point $v_1 \in [0, 4 \cdot w)^d = [0,1)^2$ is chosen, and we add the clusters $G_{v_1}^{2,\frac{1}{4}}$ (colored in green). The algorithm continues to create clusters in these manner until the entire space is clustered.*



That being said, there is a small tweak to our construction that enable us to define an ordering on the entire space without the need to check infinite number of scales. Sample a radius $r \in [\frac{1}{2}, 1]$ uniformly at random. For scale w_i, add all the points in the ball $B(\vec{0}, r \cdot w_i)$ into a special first cluster. The rest of the points will be clustered as described above, and the random ordering defined w.r.t. clustering as above. One can show that the probability that a pair of points p, q clustered together is only slightly changed. From the other hand, for every scale i such that $\xi^i \geq 2 \cdot \max\{\|p\|_2, \|q\|_2\}$, p and q will necessarily be clustered together, and thus it will be enough to check only finite number of scales (which can be only smaller than the representation length of p, q as binary strings).

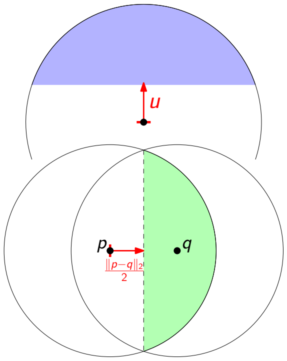
3.2 The volume of ball intersection

Consider a pair of points $p, q \in \mathbb{R}^d$ and the random partition of the space $\mathcal{C}^{w_i}$. p and q will belong to the same cluster if and only if the first center chosen in $B(p, w_i) \cup B(q, w_i)$ (that is the first ball clustering either p or q) will be in the intersection $B(p, w_i) \cap B(q, w_i)$ (that is clustering them both). As the distribution of centers in $B(p, w_i) \cup B(q, w_i)$ is uniform, the probability of p and q to be clustered together is equal to the ratio $\frac{\mathrm{Vol}_d(B(p,w_i) \cap B(q,w_i))}{\mathrm{Vol}_d(B(p,w_i) \cup B(q,w_i))}$.

Charikar *et al.* [CCG+98] analyzed this ratio (see Theorem 3.1.), as they essentially used the same partition created here. They proved the following fact:

Fact 1 ([CCG+98])**.** *For every pair of points $p, q \in \mathbb{R}^d$, the probability that p and q belong to the same clusters in $\mathcal{C}^{w_i}$ is at least $\frac{\mathrm{Vol}_d(B(p,w_i) \cap B(q,w_i))}{\mathrm{Vol}_d(B(p,w_i) \cup B(q,w_i))} \geq 1 - \sqrt{d} \cdot \frac{\|p-q\|_2}{w_i}$.*

Fact 1 is only meaningful in the regime where $\|p - q\|_2 \leq \frac{w_i}{\sqrt{d}}$. For farther p, q we will follow an analysis similar to Andoni [And09] (see Fact 2.3.1). We provide here a direct analysis to save on a second degree factor $\sqrt{d}$, that becomes significant in the important regime $t = \sqrt{d}$. A



hyperspherical cup of radius r at distance u from the origin is the set of all the points in the ball $B(\vec{0}, r)$ with the value of the first coordinate being at least u. See the blue area in the illustration on the right. The volume of the intersection of two balls of radius r, is equivalent to the volume of the union of two hyperspherical cups of radius r at distance $\frac{\|p-q\|_2}{2}$ from the origin (see illustration on the right). We provide a proof for the following technical claim in Appendix A.

Claim 1. *Denote by V_d the volume of a unit ball in $\mathbb{R}^d$. The volume of a hyperspherical cup of radius r at distance $\alpha \cdot r$ (for $\alpha \in (\frac{1}{2\sqrt{d}}, \frac{1}{2})$) from the origin is $\Omega(1) \cdot \frac{1}{\alpha \cdot \sqrt{d}} \cdot (1-\alpha^2)^{\frac{d+1}{2}} \cdot V_d \cdot r^d$.*

The following fact immediately follows:

Fact 2. *For every pair of points $p, q \in \mathbb{R}^d$ at distance $\|p-q\|_2 \in [\frac{w_i}{2\sqrt{d}}, w_i]$, the probability that p and q belong to the same clusters in $\mathcal{C}^{w_i}$ is bounded by*

$$\frac{Vol_d\left(B(p,w_i) \cap B(q,w_i)\right)}{Vol_d\left(B(p,w_i) \cup B(q,w_i)\right)} = \Omega(\frac{1}{\sqrt{d}} \cdot \frac{w_i}{\|p-q\|_2}) \cdot \left(1-(\frac{\|p-q\|_2}{2w_i})^2\right)^{\frac{d}{2}} \quad .$$

3.3 Proof of Theorem 1

We will assume here that d is large enough. Note that for constant d, $O(1)^d$ is a constant, and hence Theorem 1 follows from [CHJ20]. Further, we will assume that $t < \sqrt{d}$, as otherwise, one can plug in $t = \sqrt{d}$ in Theorem 1, and get a better stretch, with the same asymptotic number of orderings. Fix $t \in [2, \sqrt{d}]$. Pick $\gamma = c_\gamma \cdot \frac{d}{t^2} \in \mathbb{N}$, and $\epsilon = \frac{1}{c_\epsilon} \cdot \min\left\{\frac{1}{d}, \frac{\delta}{t}\right\} = \Omega\left(\frac{\delta}{d}\right)$ for large enough constants c_γ, c_ϵ, to be determined later. We begin by analyzing the success probability of a single pair in a random ordering.

A natural way to construct the LSO will be to use fresh random bits to choose cluster centers in each scale. However, it is unclear how to succeed with all possible point pairs using such an approach (and it is also computationally inferior). Instead, we will reuse the same choices (scaled) in many different weight scales. Specifically, we will create well defined orderings with finite random seeds, by choosing random centers for the scales $0, 1, \dots, \gamma-1$. Then, for a scale i such that $i \bmod \gamma = j$, we will "reuse" the centers created for the j'th scale. More formally, in order to sample an ordering $\preceq$, for every $j \in \{0, 1, \dots, \gamma-1\}$, sample centers $v_1^j, \dots, v_{\ell_j}^j \in [0, 4 \cdot w_j^d)$ for $w_j = \xi^j$ (where $\xi = \frac{28\sqrt{d+1}}{t}$). Then for every distance scale $i \in \mathbb{Z}$, such that $i = j + k \cdot \gamma$ for some $k \in \mathbb{Z}$, we will simply use the centers $\xi^{k\cdot\gamma} \cdot v_1^j, \dots, \xi^{k\cdot\gamma} \cdot v_{\ell_j}^j \in [0, 4 \cdot w_i)^d$. The ordering $\preceq$ is now well defined.

Sample an ordering $\preceq$ as above. Given two points p, q at distance $\|p-q\|_2 \leq \frac{1}{t}$, we say that the pair (p, q) is *i-successful* (w.r.t. $\preceq$) if in the clustering $\mathcal{C}^{w_i}$, the balls $B(p, \epsilon), B(q, \epsilon)$ belong to the same cluster (rather than only p, q). We say that (p, q) is *very-successful* (w.r.t. $\preceq$), if (p, q) is i-successful for every $i \geq 0$. Note that the radius ϵ does not scale with i here.

Fix $i \geq 0$, following the discussion in Section 3.1, and the triangle inequality, (p, q) is i-successful if and only if the first cluster center chosen from $B(p, \xi^i + \epsilon) \cup B(q, \xi^i + \epsilon)$ is in $B(p, \xi^i - \epsilon) \cap B(q, \xi^i - \epsilon)$.

Denote by Ψ_0 the event that (p,q) is 0-successful. The probability that (p,q) is 0-successful is thus

$$\begin{aligned}
\Pr\left[\Psi_0\right] &= \frac{\mathrm{Vol}_d\left(B(p,1-\epsilon)\cap B(q,1-\epsilon)\right)}{\mathrm{Vol}_d\left(B(p,1+\epsilon)\cup B(q,1+\epsilon)\right)}\\
&= \frac{\mathrm{Vol}_d\left(B(p,1)\cup B(q,1)\right)}{\mathrm{Vol}_d\left(B(p,1+\epsilon)\cup B(q,1+\epsilon)\right)}\cdot\frac{\mathrm{Vol}_d\left(B(p,1)\cap B(q,1)\right)}{\mathrm{Vol}_d\left(B(p,1)\cup B(q,1)\right)}\cdot\frac{\mathrm{Vol}_d\left(B(p,1-\epsilon)\cap B(q,1-\epsilon)\right)}{\mathrm{Vol}_d\left(B(p,1)\cap B(q,1)\right)}\\
&\overset{(*)}{=} (1+\epsilon)^{-d}\cdot\Omega(\frac{1}{\|p-q\|\cdot\sqrt{d}})\cdot\left(1-(\frac{\|p-q\|_2}{2})^2\right)^{\frac{d}{2}}\cdot(1-\epsilon)^d\\
&\overset{(**)}{=} \Omega(\frac{t}{\sqrt{d}})\cdot\left(1-\frac{1}{4t^2}\right)^{\frac{d}{2}}\geq\Omega(\frac{t}{\sqrt{d}})\cdot e^{-\frac{d}{2}\cdot\frac{1}{4t^2}\cdot(1+\frac{1}{2t^2})}\ ,
\end{aligned}\tag{1}$$

where the equality $^{(*)}$ follows by symmetry and Fact 2, equality $^{(**)}$ follows by our choise of ϵ and $\|p-q\|_2\leq\frac{1}{t}$, and the last equality follows by the fact $1-x\geq e^{-\frac{x}{1-x}}\geq e^{-x\cdot(1+2x)}$.

For $i\geq 1$, following the same logic as above, and using Fact 1, the probability that (p,q) is i-successful is

$$\frac{\mathrm{Vol}_d\left(B(p,\xi^i-\epsilon)\cap B(q,\xi^i-\epsilon)\right)}{\mathrm{Vol}_d\left(B(p,\xi^i+\epsilon)\cup B(q,\xi^i+\epsilon)\right)}\geq(1+\frac{\epsilon}{\xi^i})^{-d}\cdot(1-\sqrt{d}\cdot\frac{\|p-q\|_2}{\xi^i})\cdot(1-\frac{\epsilon}{\xi^i})^d>e^{-\left(\frac{2\sqrt{d}}{t\cdot\xi^i}+\frac{3\epsilon d}{\xi^i}\right)}>e^{-\frac{3\sqrt{d}}{t\cdot\xi^i}}\ ,\tag{2}$$

where the second inequality follows by the facts $1-x\geq e^{-2x}$ and $1+x\leq e^x$, and the last inequality hold for large enough c_ϵ (as $\epsilon\leq\frac{1}{c_\epsilon\cdot d}$). For $i\in[1,\gamma-1]$, all these probabilities are independent. Denote by Ψ_1 the event that (p,q) is i-successful in all $i\in[1,\gamma-1]$. Then

$$\Pr\left[\Psi_1\right]\overset{(2)}{\geq}\Pi_{i=1}^{\gamma-1}e^{-\frac{3\sqrt{d}}{t\cdot\xi^i}}=e^{-\Sigma_{i=1}^{\gamma-1}\frac{3\sqrt{d}}{t\cdot\xi^i}}\geq e^{-\frac{6\sqrt{d}}{t\cdot\xi}}=e^{-\frac{1}{2}}\geq\frac{1}{2}\ .\tag{3}$$

Denote by Ψ_2 the event that (p,q) is i-successful for all $i\geq\gamma$. By union bound (and $e^{-x}>1-x$)

$$\Pr\left[\overline{\Psi_2}\right]\leq\sum_{i\geq\gamma}1-e^{-\frac{3\sqrt{d}}{t\cdot\xi^i}}\leq\sum_{i\geq\gamma}\frac{3\sqrt{d}}{t\cdot\xi^i}\leq\frac{6\sqrt{d}}{t\cdot\xi^\gamma}\leq\frac{1}{4}\cdot\Pr\left[\Psi_0\right]\ ,\tag{4}$$

for large enough constant c_γ (recall that $\xi=\frac{12\sqrt{d}}{t}$ and $\gamma=c_\gamma\cdot\frac{d}{t^2}$). Finally, as Ψ_0 and Ψ_1 are independent, and using union bound for Ψ_2, we conclude

$$\begin{aligned}
\Pr\left[p,q\text{ are very successful}\right]&=\Pr\left[\Psi_0\wedge\Psi_1\wedge\Psi_2\right]\geq\Pr\left[\Psi_0\wedge\Psi_1\right]-\Pr\left[\overline{\Psi_2}\right]\\
&\geq\frac{1}{2}\cdot\Pr\left[\Psi_0\right]-\frac{1}{4}\Pr\left[\Psi_0\right]=\frac{1}{4}\cdot\Pr\left[\Psi_0\right]=\Omega(\frac{t}{\sqrt{d}})\cdot e^{-\frac{d}{8t^2}\cdot(1+\frac{1}{2t^2})}\ .
\end{aligned}\tag{5}$$

Denote this probability by p_{vs}. Our analysis up to this point holds for p,q such that $\|p-q\|_2\leq\frac{1}{t}$, and (p,q) is said to be very successful if it is i-successful for $i\geq 0$. In general, for p,q such that $\frac{w_{j-1}}{t}=\frac{\xi^{j-1}}{t}<\|p-q\|_2\leq\frac{\xi^j}{t}=\frac{w_j}{t}$, we say that the pair (p,q) is i-successful (w.r.t. $\preceq$) if in the clustering $\mathcal{C}^{w_i}$, the balls $B(p,\epsilon\cdot w_j),B(q,\epsilon\cdot w_j)$ belong to the same cluster. The pair (p,q) is very successful if it is i-successful for $i\geq j$ (note that j is fixed for each pair). By symmetry, for every pair (p,q), it is very successful with probability at least $\Pr\left[p,q\text{ are very successful}\right]\geq p_{\mathrm{vs}}$.

For every $i \in \{0,\dots,\gamma-1\}$, consider the box $[0,8\cdot w_i)^d$, and let $N_i \subseteq [0,8\cdot w_i)^d$ be an $\epsilon_i = \epsilon\cdot w_i$-net: a set of points such that every two are at distance at least ϵ_i, and every point in $[0,8\cdot w_i)^d$ has a net point at distance at most ϵ_i. By volume argument, as all the balls of radius $\frac{\epsilon_i}{2}$ around net points are disjoint, and contained in the box $[-w_i, 9\cdot w_i]^d$, we have that

$$\begin{aligned}|N_i| &\le \frac{\mathrm{Vol}\left([-w_i, 9\cdot w_i]^d\right)}{\mathrm{Vol}\left(B(\vec{0},\frac{\epsilon_i}{2})\right)} = \frac{(10\cdot w_i)^d}{\pi^{\frac{d}{2}}(\frac{\epsilon\cdot w_i}{2})^d/\Gamma(\frac{d}{2}+1)} \\ &= \epsilon^{-d}\cdot 2^{O(d\log d)} \le \Omega(\frac{d}{\delta})^d\cdot 2^{O(d\log d)} = 2^{O(d\log\frac{d}{\delta})}~.\end{aligned}$$

Denote by $N = \bigcup_{i=0}^{\gamma-1} N_i$ the union of all net points. Then $|N| = \sum_{i=0}^{\gamma-1}|N_i| = O(\frac{d}{t^2})\cdot 2^{O(d\log\frac{d}{\delta})} = 2^{O(d\log\frac{d}{\delta})}$. We sample i.i.d.

$$m \coloneqq p_{\mathrm{vs}}^{-1}\cdot 2\cdot\ln|N| = O\left(\frac{d^{1.5}}{t}\cdot\log\frac{d}{\delta}\cdot e^{\frac{d}{8t^2}\cdot(1+\frac{1}{2t^2})}\right)$$

orderings $\preceq_1,\preceq_2,\dots,\preceq_m$. Given $i\in\{0,\dots,\gamma-1\}$ and two net points $p,q\in N_i$, the probability that (p,q) is not very successful in any of the orderings is bounded by $(1-p_{\mathrm{vs}})^m \le e^{-2\ln|N|} = |N|^2$, while the number of such triples is smaller than $\frac{|N|^2}{2}$. Hence by union bound, we can assume that we picked a set of orderings such that for every $i\in\{0,\dots,\gamma-1\}$, and $p,q\in N_i$, (p,q) is very successful in some ordering.

Next, we argue that the created set of orderings is an $(m,\frac{2\xi\cdot t}{1-2\epsilon t})$-LSO for the entire space $\mathbb{R}^d$ (we will later improve the stretch guarantee). Consider a pair $p,q\in\mathbb{R}^d$. First assume that $p,q\in[1,7]^d$ and $\|p-q\|_2 \in \left(\frac{1}{\xi\cdot t}-\frac{2\epsilon}{\xi},\frac{1}{t}-2\epsilon\right] = \left(\frac{1-2\epsilon t}{\xi\cdot t},\frac{1-2\epsilon t}{t}\right]$. Then there are net points $p',q'\in N_0$ such that $\|p-p'\|_2,\|q-q'\|_2\le\epsilon$, and in particular $\|q'-p'\|_2\le\|p-q\|_2+2\epsilon\le\frac{1}{t}$. Let $\preceq$ be some ordering where p',q' are very successful. Hence the balls $B(p',\epsilon),B(q',\epsilon)$ are clustered together in all the hierarchical scales for $i\ge 0$. These clusters contain p and q as well. In particular, all the points between p to q in the ordering $\preceq$ belong to a single ball of radius 1. It follows that the distance between each such pair is at most $2\le\frac{2\xi\cdot t}{1-2\epsilon t}\cdot\|p-q\|_2$. We conclude that (p,q) is satisfied by the LSO.

Next consider the case where $p,q\in[w_i,7\cdot w_i]^d$ and $\|p-q\|_2\in\left((\frac{1}{\xi\cdot t}-\frac{2\epsilon}{\xi})\cdot w_i,(\frac{1}{t}-2\epsilon)\cdot w_i\right]$. If $i\in\{0,\dots,\gamma-1\}$, then the same argument as above shows that (p,q) is satisfied by the LSO (because there are close very successful net points in N_i). For general i, let $j,k\in\mathbb{Z}$ such that $i=j+k\cdot\gamma$. By symmetry, and the fact that we reused the same centers every γ scales, (p,q) is satisfied by the LSO if and only if $(\xi^{-k\gamma}\cdot p,\xi^{-k\gamma}\cdot q)$ is satisfied. But this follow from the previous case.

Finally, consider the case where $\|p-q\|_2\in\left((\frac{1}{\xi\cdot t}-\frac{2\epsilon}{\xi})\cdot w_i,(\frac{1}{t}-2\epsilon)\cdot w_i\right]$ (but not necessarily $p,q\in[w^i,7\cdot w^i]^d$). There is a vector $x\in 4w_i\cdot\mathbb{Z}^d$ such that $p+x,q+x\in[2^i,7\cdot 2^i]^d$. By the second case, there is an ordering $\preceq$ which satisfies $(p+x,q+x)$. By symmetry, the same ordering satisfies (p,q) as well.

Next, we improve the stretch guarantee to $(1+\delta)2t$. Set $\delta'=\frac{\delta}{3}$. For every $s\in 0,1,\dots,\lfloor\log_{1+\delta'}\xi\rfloor$, construct the same LSO as above, with the only change that for every i we will use a slightly shifted scale $w_i^s=\xi^i\cdot(1+\delta')^s$ (instead of $w_i=\xi^i$). Call the resulting LSO $\mathcal{L}_s$. Consider a pair (p,q), and let i,s be the unique indices such that

$$(\frac{1}{t}-2\epsilon)\cdot\xi^i\cdot(1+\delta')^{s-1} < \|p-q\|_2 \le (\frac{1}{t}-2\epsilon)\cdot\xi^i\cdot(1+\delta')^s~.$$

In the LSO $\mathcal{L}_s$, there is an ordering $\preceq$, where the pair (p,q) is i'-successful for every $i' \geq i$. In particular, by the analysis above, for every $i' \geq i$, both p,q are contained in a cluster of $\mathcal{C}^s_{i'}$. Further, in $\preceq$, all the points between p to q (and including p,q) are contained in a ball of radius $\xi^i \cdot (1+\delta')^s$, and thus are at distance at most $2 \cdot \xi^i \cdot (1+\delta')^s \leq \frac{1+\delta'}{1-2\epsilon t} \cdot 2t \cdot \|p-q\|_2$. As this holds for every pair (p,q), it follows that the union $\cup_{s=0}^{\lfloor \log_{1+\delta'} 2 \rfloor} \mathcal{L}_s$ has $\frac{1+\delta'}{1-2\epsilon t} \cdot 2t \leq \frac{1+\delta/3}{1-\delta/3} \cdot 2t \leq (1+\delta) \cdot 2t$ stretch guarantee. The total number of orderings is $\lfloor \log_{1+\delta'} \xi \rfloor \cdot m = O\left(\frac{d^{1.5}}{\delta \cdot t} \cdot \log(\frac{2\sqrt{d}}{t}) \cdot \log \frac{d}{\delta} \cdot e^{\frac{d}{8t^2} \cdot (1+\frac{1}{2t^2})}\right)$. Theorem 1 now follows by replacing t with $\frac{t}{2}$.

3.4 Generalization for ℓ_p spaces for $p \in [1,2]$.

in this subsection we generalize Theorem 1 to general ℓ_p spaces. We begin with a simple corollary for the maximal stretch regime. Afterwards, we obtain the full #orderings-stretch tradeoff for $p \in [1,2]$

Corollary 1. *The is a collection of $O(d \cdot \log d)$ orderings Σ of all the points in $\mathbb{R}^d$, such that for every $p \in [2,\infty]$, Σ is an $\left(O\left(d \cdot \log d\right), d^{1-\frac{1}{p}}\right)$-triangle LSO for d-dimensional ℓ_p space, and for every $p \in [1,2]$, Σ is an $\left(O\left(d \cdot \log d\right), d^{\frac{1}{p}}\right)$-triangle LSO for d-dimensional ℓ_p space.*

Proof. Apply Theorem 1 with parameters $\delta = \frac{1}{2}$ and $t = \frac{2}{3}\sqrt{d}$ to obtain a $\left(O\left(d \cdot \log d\right), \sqrt{d}\right)$-triangle LSO Σ for the d-dimensional Euclidean space. We argue that Σ fulfills the properties stated in Corollary 1. The following folklore fact will be crucial to our proof:

Fact 3. *For $1 \leq q \leq p$, and every vector $x \in \mathbb{R}^d$, it holds that $\|x\|_p \leq \|x\|_q \leq d^{\frac{1}{q}-\frac{1}{p}} \|x\|_p$.*

Consider a pair of points $x, y \in \mathbb{R}^d$. There is an ordering $\sigma \in \Sigma$ such that w.l.o.g. $x \prec_\sigma y$, and for every $a, b \in \mathbb{R}^d$ such that $x \preceq_\sigma a \preceq_\sigma b \preceq_\sigma y$ it holds that $\|a-b\|_2 \leq d^{\frac{1}{2}} \cdot \|x-y\|_2$. Then for every $p \in [2,\infty]$ and $a, b \in \mathbb{R}^d$ such that $x \preceq_\sigma a \preceq_\sigma b \preceq_\sigma y$, by Fact 3 it holds that

$$\|a-b\|_p \leq \|a-b\|_2 \leq d^{\frac{1}{2}} \cdot \|x-y\|_2 \leq d^{1-\frac{1}{p}} \cdot \|x-y\|_p \ .$$

Similarly, for $p \in [1,2]$, and $a, b \in \mathbb{R}^d$ such that $x \preceq_\sigma a \preceq_\sigma b \preceq_\sigma y$ it holds that

$$\|a-b\|_p \leq d^{\frac{1}{p}-\frac{1}{2}} \cdot \|a-b\|_2 \leq d^{\frac{1}{p}} \cdot \|x-y\|_2 \leq d^{\frac{1}{p}} \cdot \|x-y\|_p \ .$$

The corollary now follows. □

Next we prove Theorem 2, restated bellow for convenience.

Theorem 2. *For every $p \in [1,2]$, $t \in [5, d^{\frac{1}{p}}]$ and $d \geq 1$, the d-dimensional ℓ_p space admits $\left(e^{O(\frac{d}{t^p})} \cdot \tilde{O}(d), t\right)$-triangle LSO.*

Proof. Denote by $B_p(x,r) = \{y \mid \|x-y\|_p \leq r\}$ the ℓ_p ball around x of radius r. We will repeat the same algorithm from Section 3.1 with small changes. First, instead of using standard Euclidean balls we will use ℓ_p balls. Second, the gap between consecutive scales will be $\xi_p = \frac{36d}{t}$ instead of $\xi = \frac{12\sqrt{d}}{t}$ (this introduces additional poly(d) factor to the number of orderings). Thus we will obtain random clustering $\mathcal{C}_p^{w_i}$ in each scale w_i, which will induce an LSO. We begin with analyzing the probability that nearby points will be clustered together. This boils down to the volume of the intersection between two ℓ_p balls. The proof of the following lemma follows the lines of a similar proof in [Ngu13], and is differed to Appendix A.

Lemma 1. *For $p \in [1,2]$, and every pair of points $x,y \in \mathbb{R}^d$ at distance $\|x-y\|_p = \frac{1}{t} \in [d^{-\frac{1}{p}}, \frac{1}{3}] \cdot w_i$, the probability that x and y belong to the same clusters in $\mathcal{C}_p^{w_i}$ is bounded by*

$$\frac{Vol_d\left(B_p(x,w_i) \cap B_p(y,w_i)\right)}{Vol_d\left(B_p(x,w_i) \cup B_p(y,w_i)\right)} = e^{-\Omega(\frac{d}{t^p})}$$

The following lemma is quite straightforward, and far from being tight. However, the loss will not be significant in our analysis. The proof is differed to Appendix A.

Observation 1. *For $p \in [1,2]$, and every pair of points $x,y \in \mathbb{R}^d$ at distance $\|x-y\|_p \le \frac{w_i}{4d}$, the probability that x and y belong to the same clusters in $\mathcal{C}_p^{w_i}$ is bounded by*

$$\frac{Vol_d\left(B_p(x,w_i) \cap B_p(y,w_i)\right)}{Vol_d\left(B_p(x,w_i) \cup B_p(y,w_i)\right)} \ge 1 - 4d \cdot \frac{\|x-y\|_p}{w_i} \ .$$

The rest of the analysis follows the exact same lines as the proof of Theorem 1, while replacing the usage of Fact 1 and Fact 2 by Observation 1 and Lemma 1 respectively, and using a larger gap $\xi_p = \frac{36d}{t}$ between consecutive scales in a single LSO. The other parameters, $\gamma = c_\gamma \cdot \frac{d}{t^p} \in \mathbb{N}$, and $\epsilon = \frac{1}{c_\epsilon} \cdot \min\left\{\frac{1}{d}, \frac{\delta}{t}\right\} = \Omega\left(\frac{\delta}{d}\right)$ remain the same, while we fix $\delta = \frac{1}{4}$.

Specifically, consider a pair x,y at distance $\|x-y\|_p \le \frac{1}{t}$. The definition of i-successful, very successful, and Ψ_0, Ψ_1, Ψ_2 remains the same. Following Equation (1) and using Lemma 1,

$$\Pr\left[\Psi_0\right] = (1+\epsilon)^{-d} \cdot e^{-\Omega(\frac{d}{t^p})} \cdot (1-\epsilon)^d = e^{-\Omega(\frac{d}{t^p})} \ .$$

Following Equation (2) and using Observation 1, the probability that (x,y) is i-successful is

$$\frac{\mathrm{Vol}_d\left(B_p(x,\xi_p^i - \epsilon) \cap B_p(y,\xi_p^i - \epsilon)\right)}{\mathrm{Vol}_d\left(B_p(x,\xi_p^i + \epsilon) \cup B_p(y,\xi_p^i + \epsilon)\right)} \ge (1+\frac{\epsilon}{\xi_p^i})^{-d} \cdot (1 - 4d \cdot \frac{\|p-q\|_p}{\xi_p^i}) \cdot (1 - \frac{\epsilon}{\xi_p^i})^d > e^{-\left(\frac{8d}{t \cdot \xi_p^i} + \frac{3\epsilon d}{\xi_p^i}\right)} > e^{-\frac{9d}{t \cdot \xi_p^i}} \ .$$

Hence by independence (following Equation (3)), and by union bound (following Equation (4)) we have

$$\Pr\left[\Psi_1\right] \ge \Pi_{i=1}^{\gamma-1} e^{-\frac{9d}{t \cdot \xi_p^i}} = e^{-\sum_{i=1}^{\gamma-1} \frac{9d}{t \cdot \xi_p^i}} \ge e^{-\frac{18d}{t \cdot \xi_p}} = e^{-\frac{1}{2}} \ge \frac{1}{2} \ .$$

$$\Pr\left[\overline{\Psi_2}\right] \le \sum_{i \ge \gamma} 1 - e^{-\frac{9d}{t \cdot \xi_p^i}} \le \sum_{i \ge \gamma} \frac{9d}{t \cdot \xi_p^i} \le \frac{18d}{t \cdot \xi_p^\gamma} \le \frac{1}{4} \cdot \Pr\left[\Psi_0\right] \ .$$

Following Equation (5), the probability that (x,y) is very successful is at least $p_{\text{vs}} \ge \frac{1}{4} \cdot \Pr\left[\Psi_0\right] = e^{-\Omega(\frac{d}{t^p})}$. We pick $N_i \subseteq [0, 8 \cdot w_i)^d$ an $\epsilon_i = \epsilon \cdot w_i$-net (w.r.t. $\|\cdot\|_p$) of size $2^{O(d \log \frac{d}{\delta})}$, and denote $N = \bigcup_{i=0}^{\gamma-1} N_i$. Sampling $m_p \coloneqq p_{\text{vs}}^{-1} \cdot 2 \cdot \ln|N| = d \cdot \log d \cdot e^{O(\frac{d}{t^p})}$ i.i.d. orderings we can insure that every pair of points $x,y \in N_i$ is very successful in some ordering. Following the same arguments as in the proof of Theorem 1 (triangle inequality), we conclude that this ordering is an $(m, \frac{2\xi_p \cdot t}{1-2\epsilon t})$-LSO for the entire space $(\mathbb{R}^d, \|\cdot\|_p)$. By adding $\log \xi_p = O(\log \frac{2d}{t})$ different shifts of this construction, we obtain an $(m_p \cdot \log \xi_p, (1+\delta) \cdot 2t) = \left(d \cdot \log d \cdot \log \frac{2d}{t} \cdot e^{O(\frac{d}{t^p})}, (1+\delta) \cdot 2t\right)$-triangle LSO. Theorem 2 now follows by replacing t with $\frac{2t}{5}$.

□

4 LSO for high dimensional doubling spaces

This section is devoted to proving the following theorem (restated for convenience):

Theorem 3. *Given a metric space (X, d_X) with doubling dimension d, and parameter $t \in [\Omega(1), d]$, X admits $\left(2^{O(d/t)} \cdot d \cdot \log^2 t, t\right)$-triangle LSO.*

We begin by introducing the notion of *padded partition cover scheme.* This notion is closely related to padded decompositions and sparse covers (see e.g. [Fil19] and references therein).

Definition 5 (Padded Partition Cover Scheme)**.** *A partition $\mathcal{P}$ of a metric space (X, d_X) is Δ-bounded if the diameter of every cluster $C \in \mathcal{P}$ is at most Δ. A collection of partitions $\mathcal{P}_1, \dots, \mathcal{P}_s$ is (τ, ρ, Δ)-padded partition cover if (a) $s \le \tau$, (b) every partition $\mathcal{P}_i$ is Δ-bounded, and (c) for every point x, there is a cluster C in one of the partitions $\mathcal{P}_i$ such that $B(x, \frac{\Delta}{\rho}) \subseteq C$.*
*A space (X, d_X) admits a (τ, ρ)-*padded partition cover scheme *if for every Δ, it admits a (τ, ρ, Δ)-padded partition cover.*

Filtser [Fil19] implicitly constructed padded partition cover scheme for doubling metrics. Explicitly [Fil19] only argued that the construction is sparse cover, [9] however it is actually a padded partition cover scheme.

Theorem 5 ([Fil19])**.** *Let (X, d_X) be a metric space with doubling dimension d and parameter $t = \Omega(1)$. Then G admits a $\left(t, O(2^{O(d/t)} \cdot d \cdot \log t)\right)$-padded partition cover scheme.*

An *ultrametric* (X, d) is a metric space satisfying a strong form of the triangle inequality, that is, for all $x, y, z \in X$, $d(x, z) \le \max\{d(x, y), d(y, z)\}$. A related notion is *hierarchical well-separated tree* (HST).

Definition 6 (HST)**.** *A metric (X, d_X) is an hierarchical well-separated tree (HST) if there exists a bijection φ from X to leaves of a rooted tree T in which:*

1. *Each node $v \in T$ is associated with a label Γ_v such that $\Gamma_v = 0$ if v is a leaf and $\Gamma_v \ge \Gamma_u$ if v is an internal node and u is any child of v.*
2. *$d_X(x, y) = \Gamma_{\mathsf{lca}(\varphi(x), \varphi(y))}$ where $\mathsf{lca}(u, v)$ is the least common ancestor of any two given nodes u, v in T.*

It is well known that any ultrametric an HST are equivalent definitions (see [BLMN05b]).

Ultrametric cover. Filtser and Hung [FL22] introduced the notion of ultrametric cover. Consider a metric space (X, d_X), a distance measure d_Y is said to be dominating if $\forall x, y \in X$, $d_X(x, y) \le d_Y(x, y)$.

Definition 7 (Ultrametric Cover)**.** *A (τ, ρ)-*ultrametric cover *for a space (X, d) is a collection of at most τ dominating ultrametrics $\mathcal{U} = \{(U_i, d_{U_i})\}_{i=1}^{\tau}$ over X, such that for every $x, y \in X$, $d_X(x, y) \le \min_{1 \le i \le \tau} d_{U_i}(x, y) \le \rho \cdot d_X(x, y)$.*

In [FL22] it was shown that given an ultrametric cover, one can construct a triangle-LSO.

[9] (τ, ρ, Δ)- Sparse cover is a collection $\mathcal{C}$ of non-disjoint clusters of diameter at most Δ, such that every vertex belongs to at most τ clusters, and every ball of radius Δ/ρ is contained in some cluster. Note that the union of all the partitions in a (τ, ρ, Δ)-padded partition cover is a (τ, ρ, Δ)-cover (but not necessarily vice versa).

Lemma 2. *[[FL22]] If a metric (X, d_X) admits a (τ,ρ)-ultrametric cover $\mathcal{U}$, then it has a (τ,ρ)-triangle-LSO.*

Thus the main technical part of this section is the construction of ultrametric covers for doubling spaces. We will prove a reduction from padded partition cover to ultrametric cover. In essence, ultrametric is simply a hierarchical partition. Thus, this reduction takes unrelated partitions in all possible distance scales, and combines them into hierarchical/laminar partition. Reductions similar in spirit were constructed in the context of the Steiner point removal problem [Fil20b], stochastic Steiner point removal [EGK+14], universal Steiner tree [JLN+05], as well as for ultrametric covers [FL22], and others. We follow here a bottom-up approach, where the ratio between consecutive scales in a single hierarchical partition (a.k.a. ultrametric) is $O(\frac{\rho}{\epsilon})$. When constructing the next level in the hierarchical partition, we take partitions from a padded partition cover of the current scale, and slightly "round" them around the "borders" so that no previously created cluster will be divided (see Figure 3). The argument is that due to the large ratio between consecutive scales, the effects of this rounding are marginal. The proof appears bellow

Lemma 3. *Suppose that a metric space (X, d_X) admits a (τ,ρ)-padded partition cover scheme. Then for every $\epsilon\in(0,\frac{1}{2})$, (X,d_X) admits an $\big(O(\frac{\tau}{\epsilon}\log\frac{\rho}{\epsilon}),\rho(1+\epsilon)\big)$-ultrametric cover.*

By plugging Theorem 5 ([Fil19]) into Lemma 3 (with $\epsilon=\frac{1}{4}$ and $t'=\frac{t}{1+\epsilon}$) we obtain that every metric space with doubling dimension d admits an $\big(2^{O(d/t)}\cdot d\cdot\log^2 t, t\big)$-ultrametric cover. Now, Theorem 3 follows by Lemma 2 ([FL22]).

Proof of Lemma 3. Assume w.l.o.g. that the minimal pairwise distance in X is 1, while the maximal pairwise distance is Φ. Fix $\epsilon\in(0,\frac{1}{4})$ and $c\geq 1$ to be determined later. For $i\geq 0$, set $\Delta_i=c\cdot(\frac{4\rho}{\epsilon})^i$, and let $\mathbb{P}_i=\{\mathcal{P}_1^i,\dots,\mathcal{P}_\tau^i\}$ be a (τ,ρ,Δ_i)-padded partition cover (we assume that $\mathbb{P}_i$ has exactly τ partitions; we can enforce this assumption by duplicating partitions if necessary). Fix some j, let $\mathcal{P}_j^{-1}$ be the partition where each vertex is a singleton, and consider $\{\mathcal{P}_j^i\}_{i\geq -1}$. We will inductively define a new set of partitions, enforcing it to be a laminar system. The basic idea is to produce a tree of partitions where the lower level is a refinement of the higher level, and we do so by grouping a cluster at a lower level to one of the clusters at a higher level separating it.

The lowest level $\mathcal{P}_j^{-1}$ where each set in the partition is a singleton, stays as-is. Inductively, for any $i\geq 0$, after constructing $\tilde{\mathcal{P}}_j^{i-1}$ from $\mathcal{P}_j^{i-1}$, we will construct $\tilde{\mathcal{P}}_j^i$ from $\mathcal{P}_j^i$ using $\tilde{\mathcal{P}}_j^{i-1}$. Let $\mathcal{P}_j^i=\big\{C_1,\dots,C_\phi\big\}$ be all sets in the partition $\mathcal{P}_j^i$. For each $q\in[1,\phi]$, let $Y_q=X\setminus\cup_{a<q}\tilde{C}_a$ be the set of unclustered points (w.r.t. level i). Let $C'_q=C_q\cap Y_q$, and let $S_{C'_q}=\big\{C\in\tilde{\mathcal{P}}_j^{i-1}\mid C\cap C'_q\neq\emptyset\big\}$ be the set of new level-$(i-1)$ clusters intersecting C'_q. We set $\tilde{C}_q=\cup S_{C'_q}$ to be the union of these clusters and continue iteratively. See Figure 3 for illustration. Clearly, $\tilde{\mathcal{P}}_j^{i-1}$ is a refinement of $\tilde{\mathcal{P}}_j^i$. We conclude that $\big\{\tilde{\mathcal{P}}_j^i\big\}_{i\geq -1}$ is a laminar hierarchical set of partitions that refine each other.

We argue by induction that $\tilde{\mathcal{P}}_j^i$ has diameter $\Delta_i(1+\epsilon)$. Consider $\tilde{C}_q\in\tilde{\mathcal{P}}_j^i$; it consists of $C'_q\subseteq C_q\in\mathcal{P}_j^i$ and of clusters in $\tilde{\mathcal{P}}_j^{i-1}$ intersecting C'_q. As the diameter of C'_q is bounded by $\mathrm{diam}(C_q)\leq\Delta_i$, and by the induction hypothesis, the diameter of each cluster $C\in\tilde{\mathcal{P}}_j^{i-1}$ is bounded by $(1+\epsilon)\Delta_{i-1}$, we conclude that the diameter of $\tilde{C}_q$ is bounded by

$$\Delta_i+2\cdot(1+\epsilon)\Delta_{i-1}=\Delta_i\left(1+\frac{2(1+\epsilon)}{4\rho}\cdot\epsilon\right)\leq\Delta_i(1+\epsilon)\ ,$$

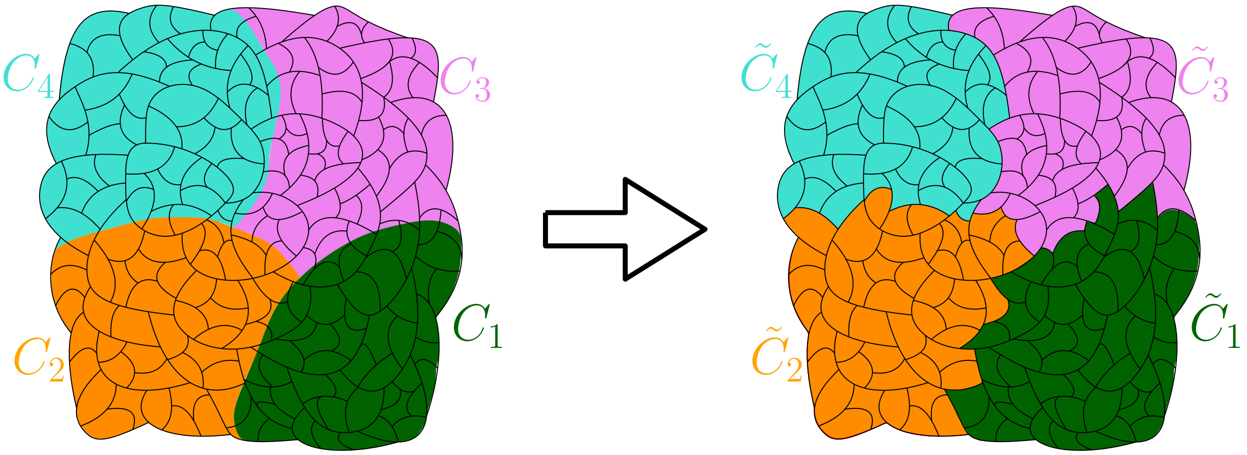


Figure 3: Illustration of the construction of the partition $\tilde{\mathcal{P}}^i_j$ given $\mathcal{P}^i_j$ and $\tilde{\mathcal{P}}^{i-1}_j$. The black lines in both the left and right parts of the figure border clusters in $\tilde{\mathcal{P}}^{i-1}_j$. On the left illustrated the partition $\mathcal{P}^i_j = \{C_1, C_2, C_3, C_4\}$, where different clusters colored by different colors. On the right illustrated the modified partition $\mathcal{P}^i_j = \{\tilde{C}_1, \tilde{C}_2, \tilde{C}_3, \tilde{C}_4\}$. $\tilde{C}_1$ contains all the clusters in $\tilde{\mathcal{P}}^{i-1}_j$ intersecting C_1. C_2 contains all the clusters in $\tilde{\mathcal{P}}^{i-1}_j \setminus \tilde{C}_1$ intersecting C_2, and so on.

since $\rho \ge 1$ and $\epsilon < 1$.

Next we argue that $\tilde{\mathbb{P}}_i = \{\tilde{\mathcal{P}}^i_1, \dots, \tilde{\mathcal{P}}^i_\tau\}$ is a $(\tau, (1+\epsilon)^2\rho, (1+\epsilon)\Delta)$-padded partition cover. Observe that it contains τ partitions, and we have shown that all the clusters have diameter at most $(1+\epsilon)\Delta$. Thus, it remains to prove the padding property. Consider a vertex $v \in X$, there is some index j such that $B(x, \frac{\Delta_i}{\rho}) \subseteq C_i \in \mathcal{P}^i_j$. We argue that $B(x, \frac{(1+\epsilon)\Delta_i}{(1+\epsilon)^2\rho}) \subseteq \tilde{C}_i \in \mathcal{P}^i_j$. Note that if a cluster $\tilde{C}_{i-1}$ is not contained in $\tilde{C}_i$, then it must contain a vertex out of C_i. In other words, every vertex $y \notin \tilde{C}_i$ belongs to a cluster $\tilde{C}_y \in \mathcal{P}^{i-1}_j$ containing vertex $z \notin C_i$. Necessarily $d_X(x,z) > \frac{\Delta_i}{\rho}$. Using the triangle inequality

$$d_X(x,y) \ge d_X(x,z) - d_X(z,y) > \frac{\Delta_i}{\rho} - (1+\epsilon)\Delta_{i-1} = \frac{\Delta_i}{\rho}\left(1 - \frac{(1+\epsilon)\epsilon}{4}\right) \ge \frac{\Delta_i}{\rho(1+\epsilon)} = \frac{(1+\epsilon)\Delta_i}{(1+\epsilon)^2\rho} .$$

If follows that $B(x, \frac{\Delta_i(1+\epsilon)}{\rho(1+\epsilon)^2}) \subseteq \tilde{C}_i \in \tilde{\mathcal{P}}^i_j$.

Finally, we construct an ultrametric cover. Fix an index $j \in [1,\tau]$; we construct an HST U_j as follows. Leaves of U_j bijectively correspond to points in X and have label 0. For each $i \in [0, I]$ where $I = \lceil \log_{4\rho/\epsilon} \Phi/c \rceil$, internal nodes at level i bijectively correspond to the clusters $\tilde{\mathcal{P}}^i_j$ (leaves of U_j is at level -1), and have label $(1+\epsilon)\Delta_i$. There is an edge from each node corresponding to a cluster $\tilde{C}_{i-1} \in \tilde{\mathcal{P}}^{i-1}_j$ to the node corresponding the unique cluster $\tilde{C}_i \in \tilde{\mathcal{P}}^i_j$ containing $\tilde{C}_{i-1}$. The root of U_j is the unique single cluster in $\tilde{\mathcal{P}}^I_j$. Clearly, the ultrametric cover $\{U_j\}_{j=1}^{\tau}$ is dominating.

To bound the stretch, we will construct such an ultrametric cover with $c = (1+\epsilon)^l$ for every $l \in [0, \lfloor \log_{1+\epsilon} \frac{4\rho}{\epsilon} \rfloor]$. The final ultrametric cover will be a union of these $O(\log_{1+\epsilon} \frac{4\rho}{\epsilon})$ ultrametric covers. Clearly, their cardinality is bounded by $\tau \cdot O(\log_{1+\epsilon} \frac{4\rho}{\epsilon}) = O(\frac{\tau}{\epsilon} \log \frac{\rho}{\epsilon})$.

Consider a pair $x, y \in X$. Let $l \in [0, \lfloor \log_{1+\epsilon} \frac{4\rho}{\epsilon} \rfloor]$, and $i \ge 0$ be the unique indices such that $(1+\epsilon)^{l-1}(\frac{4\rho}{\epsilon})^i \le (1+\epsilon)\rho \cdot d_X(x,y) \le (1+\epsilon)^l(\frac{4\rho}{\epsilon})^i$. For the ultrametric cover constructed w.r.t. $c = (1+\epsilon)^l$, $\Delta_i = c \cdot (\frac{4\rho}{\epsilon})^i$, and hence there is some index j, and a cluster $\tilde{C}_i \in \tilde{\mathcal{P}}^i_j$ such that

$B(x, \frac{\Delta_i(1+\epsilon)}{\rho(1+\epsilon)^2}) \subseteq \tilde{C}_i \in \tilde{\mathcal{P}}^i_j$. It holds that

$$d_X(x,y) = \frac{(1+\epsilon)\rho \cdot d_X(x,y)}{(1+\epsilon)\rho} \le \frac{(1+\epsilon)^l \cdot (\frac{4\rho}{\epsilon})^i}{(1+\epsilon)\rho} = \frac{c \cdot (\frac{4\rho}{\epsilon})^i}{(1+\epsilon)\rho} = \frac{(1+\epsilon)\Delta_i}{(1+\epsilon)^2\rho} .$$

We conclude that in U_j, both x, y are decedents of a node with label $(1+\epsilon)\Delta_i \le (1+\epsilon)^2\rho \cdot d_X(x,y)$; the stretch guarantee follows.

In summary, we have constructed an $\big(O(\frac{\tau}{\epsilon}\log\frac{\rho}{\epsilon}), \rho(1+\epsilon)^2\big)$-ultrametric cover. The lemma follows by rescaling ϵ. □

Remark 1. *Our construction of triangle-LSO's for doubling metrics is much simpler than for the case of Euclidean space. One may ask, why did we went through an elaborate direct proof for the Euclidean case, instead of simply constructing padded partition cover and using Lemma 3? Interestingly, padded covers (as well as padded partitions [Fil19]) in Euclidean spaces have similar tradeoff to general doubling spaces: padding parameter t requires $\approx 2^{O(\frac{d}{t})}$ partitions. The crucial structural property we use here, is that if one wishes to insure that every pair at distance $\frac{\Delta}{t}$ is clustered together (instead of the entire ball $B(x,\frac{\Delta}{t})$), then in Euclidean space $\approx 2^{O(\frac{d}{t^2})}$ partitions are enough, a significant improvement!*

5 Labeled Nearest Neighbor Search

In this section we construct our labeled NNS data structures for various metric spaces. See Table 2 for a summary of our results. In Section 5.2 we prove a meta Theorem 7 turning rooted LSO into labeled NNS. We conclude efficient labeled NNS for planar and fixed minor free graphs (Corollary 2), treewidth graphs (Corollary 3), and metric spaces with small correlation dimension (Corollary 4). Then, in Section 5.3 we prove a meta Theorem 7 turning rooted LSO into labeled NNS. We conclude a labeled NNS for graphs with relatively large doubling dimension (Corollary 5), and for general metric spaces (Corollary 6). Later in Section 5.4 we directly construct labeled NNS for general metrics to obtain a very small query time (Theorem 12). We then focus on the $O(\log N)$-stretch regime, and construct very efficient labeled NNS structures, under various relaxations (Theorem 14 and Corollaries 7, 9, 8, 10), see Table 7 for a summary. Finally, in Section 5.5 we show that the classic information theoretic lower bound holds for labeled NNS as well (Theorem 15).

5.1 Data Structures preliminaries

In our labeled NNS construction, we will be using the following data structure by Willard [Wil83]:

Theorem 6 (Y-fast trie, [Wil83])**.** *Given a subset $S \subseteq [N]$ of n elements, there is a data structure with space $O(n)$, and such that given a query $q \in [N]$, returns in $O(\log\log N)$ time the predecessor and successor of q in S. Further, the data structure supports insertion and deletion of elements in S in amortized $O(\log\log N)$ time.*

We will use the Y-fast trie to obtain additional information:

Observation 2. *When constructing the Y-fast trie, one can also answer minimum element in S queries in $O(1)$ time, while all the other parameters remain unchanged.*

Proof. For simplicity, we will assume that the data structure works for the elements $\{0,\dots,N\}$ instead of $\{1,\dots,N\}$, and that the element 0 is never part of N. Such assumption could be insured by building the data structure for $[N+1]$, and shifting all the elements one position to the right.

When constructing the data structure in the first time we simply store the minimum element m_S. On an insert operation of an element x, we check whether $x < m_S$, if this is the case we update $m_S \leftarrow x$ (otherwise do nothing). On a remove operation of an element x, we check whether $x = m_S$. If no, do nothing. Else, we query the successor of 0, and set it to be the new m_S. Note that the total removal time is still $O(\log\log N)$ amortized time. Clearly the other parameters (space, predecessor/successor queries) remain the same, while we can answer minimum element queries in $O(1)$ time. □

5.2 Labeled NNS From Rooted LSO

In this subsection we prove a meta theorem, turning rooted-LSO's into labeled NNS. As a corollaries we obtain labeled NNS for planar graphs, fixed minor free graphs, treewidth graphs, and for metric spaces with small correlation dimension.

Theorem 7 (Meta theorem: rooted LSO to labeled NNS)**.** *Consider a metric space (X, d_X) admitting a (τ,ρ)-rooted NNS. Than there is a dynamic labeled ρ-NNS with label size $O(\tau)$, space $O(n\cdot\tau)$, query time $O(\tau)$, and deletion/insertion in $O(\tau\cdot\log\log N)$ amortized time.*

Proof. Let Σ be the promised collection of orderings of the rooted LSO. For every point $y\in X$, let $\Sigma_y\subseteq\Sigma$ be the subset of orderings y belongs to. We construct a label ℓ_y for y as follows: for every $\sigma\in\Sigma_y$ store the name of σ, the position of y in the order σ, and the distance $d_X(y,x_\sigma)$ from y to the first vertex x_σ in σ. Note that the label size consist of $O(|\Sigma_y|) = O(\tau)$ words.

We will use the Y-fast trie from Theorem 6, together with Observation 2. Consider a subset $P\subseteq X$ of n points. To construct the labeled NNS, we will first store all the labels of P points. Next, for every ordering $\sigma\in\Sigma$ that contains at least one point from P, we will construct a Y-fast trie with the inputs being the positions in σ of P points. Each point in the Y-fast trie will be linked to the relevant label. This finishes the data structure. Note that the total space is $O(n\cdot\tau)$. Moreover, we can add/remove a point by simply adding/removing its label, and updating the Y-fast tries of all the orderings it belongs to. Such an update will take $O(\tau\cdot\log\log N)$ amortized time.

Given a query $q\in X$ with label ℓ_q, we go over all the orderings Σ_q containing q. For each $\sigma\in\Sigma_q$ we use the Y-fast trie to obtain the minimal element y_σ in the order σ (among the points in P). Denote by x_σ the first element in the order (globally w.r.t. X). Our answer will be the element y_σ minimizing $d_X(q,x_\sigma)+d_X(x_\sigma,y_\sigma)$ over all (non empty) $\sigma\in\Sigma_q$. Note that extracting the minimal element and computing the sum of distances will take us $O(1)$ time. Thus the total query time is $O(\tau)$.

Finally, we argue that the returned point is a ρ approximation to the nearest neighbor. Let $p\in P$ be the closest point to q. Then there is an ordering $\sigma'\in\Sigma_q$ such that $d_X(q,x_{\sigma'})+d_X(x_{\sigma'},p)\le \rho\cdot d_X(q,p)$. Note that $y_{\sigma'}\preceq_{\sigma'} p$, thus $d_X(x_\sigma,y_{\sigma'})\le d_X(x_\sigma,p)$. We conclude

$$\begin{aligned}\min_{\sigma\in\Sigma_q}\left\{d_X(q,x_\sigma)+d_X(x_\sigma,y_\sigma)\right\} &\le d_X(q,x_{\sigma'})+d_X(x_{\sigma'},y_{\sigma'})\\ &\le d_X(q,x_{\sigma'})+d_X(x_{\sigma'},p)\quad\le(1+\epsilon)\cdot d_X(q,p)\ .\end{aligned}$$

□

Remark 2. *If one wishes to obtain a worst case update time (instead of amortized) one can replace the Y-fast tries with heaps. All the parameters will remain the same, other that the deletion/insertion that will become $O(\tau \log n)$ (worst case).*

By plugging in the rooted-LSO's of [FL22] (see Table 1) into the Theorem 7 we conclude:

Corollary 2. *For every $\epsilon \in (0.\frac{1}{2})$, planar graphs (and fixed minor free graphs) admit labeled $(1+\epsilon)$-NNS with $O(\frac{1}{\epsilon}\log^2 N)$ label size, space $n \cdot O(\frac{1}{\epsilon}\log^2 N)$, query time $O(\frac{1}{\epsilon}\log^2 N)$, and deletion/insertion $O(\frac{1}{\epsilon}\log^2 N \log\log N)$ amortized time.*

Corollary 3. *Every treewidth k graph admits an (exact) labeled 1-NNS with label size $O(k \log N)$, space $O(n \cdot k \log N)$, query time $O(k \log N)$, and deletion/insertion in $O(k \log N \log\log N)$ amortized time.*

5.2.1 Correlation dimension

Correlation dimension was introduced by Chan and Gupta [CG12] as a generalization of doubling dimension. While in doubling dimension the defining property is local (every ball of radius r can be covered by at most 2^d balls of radius $\frac{r}{2}$), in correlation dimension this property holds only on average. In particular, a metric might have a constant correlation dimension, while containing a $\sqrt{N}$-size clique.

Given a metric space (X, d_X), an r-net is a subset $N \subseteq X$ such that every pair of net points is at distance at least r: $\min_{x,y \in N} d_X(x,y) \geq r$, and every point is at distance at most r from some net point: $\max_{x \in X} \min_{y \in N} d_X(x,y) \leq r$. (X, d_X) is said to have *correlation dimension* k if for every net N and $s > 0$, it holds that

$$\sum_{x \in N} |N \cap B_X(x, 2s)| \leq 2^k \cdot \sum_{x \in N} |N \cap B_X(x, s)| \ .$$

Chan and Gupta [CG12] (generalizing the embedding of [Tal04]) showed that every N-point metric space (X, d_X) with correlation dimension k embeds into a graph with treewidth $\tilde{O}_{k,\epsilon}(\sqrt{N})$ and multiplicative distortion $1+\epsilon$. Specifically, there is a map f from X into the vertices of a graph G, where G has treewidth $\tilde{O}(\sqrt{N}) \cdot \alpha(k,\epsilon)$ (α being an arbitrary function of ϵ, k), such that for every $x, y \in X$, $d_X(x,y) \leq d_G(f(x), f(y)) \leq (1+\epsilon) \cdot d_X(x,y)$. Using this embedding, and Corollary 3, we conclude:

Corollary 4. *Every graph with correlation dimension k admit labeled $(1+\epsilon)$-NNS with label size $\tilde{O}_{k,\epsilon}(\sqrt{N})$, space $\tilde{O}_{k,\epsilon}(n \cdot \sqrt{N})$, query time $\tilde{O}_{k,\epsilon}(\sqrt{N})$, and deletion/insertion in $\tilde{O}_{k,\epsilon}(\sqrt{N})$ time.*

Note that by our lower bound (Theorem 15), there is a $\sqrt{N}$-vertex graph G such that every labeled t-NNS for $t < 3$ has label size $\tilde{\Omega}(\sqrt{N})$. There is an N-vertex graph H with constant correlation dimension, containing G as a subgraph, and such that the induced shortest path metric on G vertices remains the same. It follows that every t-labeled NNS for $t < 3$ for H has label size $\tilde{\Omega}(\sqrt{N})$. It follows that Corollary 4 is tight up to second order terms.

5.3 Labeled NNS From Triangle LSO

In this subsection we prove a meta theorem, turning triangle-LSO's into labeled NNS:

Theorem 8 (Meta theorem: triangle LSO to labeled NNS)**.** *Consider a metric space (X,d_X) admitting a (τ,ρ)-triangle NNS. Then there is a dynamic labeled 2ρ-NNS with label size $O(\tau\cdot\log N)$, space $O(n\cdot\tau\cdot\log N)$, query time $O(\tau\cdot\log\log N)$, and deletion/insertion in $O(\tau\cdot\log\log N)$ amortized time.*

Proof. Let Σ be the promised collection of orderings of the triangle LSO. For every ordering $\sigma\in\Sigma$, we form an unweighted path graph P_σ with vertex set X and the order of vertices along the path is σ. We construct a 2-hop 1 spanner H_σ for P^σ_n using Theorem 4. For every point $y\in X$, let $E_{\sigma,y}$ be the collection of edges y is responsible for. We construct a label ℓ_y for a point $y\in X$ as follows: for every $\sigma\in\Sigma$ store the position of y in the order σ, and all the edges in $E_{\sigma,y}$ together with their weight (i.e. for $\{x,y\}\in E_{\sigma,y}$ store $(\{x,y\},d_X(x,y))$). The label size is thus $O(\tau\cdot\log N)$ words.

Consider a subset $P\subseteq X$ of n points. To construct the labeled NNS, we will first store all the labels of P points. Next, for every ordering $\sigma\in\Sigma$, we will construct a Y-fast trie (Theorem 6) with the inputs being the positions in σ of P points. Each point in the Y-fast trie will be linked to the relevant label. This finishes the data structure. Note that the total space is $O(n\cdot\tau\cdot\log N)$. Moreover, we can add/remove a point by simply adding/removing its label, and updating the Y-fast tries of all the orderings it belongs to. Such an update will take $O(\tau\cdot\log\log N)$ amortized time (we assume that just coping the label itself takes $O(1)$ times as we can store a link. But to insert in to the Y-fast tries, takes $O(\log\log N)$ per ordering).

Given a query $q\in X$ with its label ℓ_q, we go over all the orderings. For each $\sigma\in\Sigma$ we use the Y-fast trie to obtain predecessor $y_{\sigma,p}$ and successor $y_{\sigma,s}$ of q w.r.t. σ in $O(\log\log n)$ time. By knowing the position of q and $y_{\sigma,s}$ in σ, we can compute in $O(1)$ time a point $\hat{y}_{\sigma,s}$ such that $q\prec_\sigma\hat{y}_{\sigma,s}\prec_\sigma y_{\sigma,s}$, and $\{q,\hat{y}_{\sigma,s}\}\in E_{\sigma,y}$, $\{\hat{y}_{\sigma,s},y_{\sigma,s}\}\in E_{\sigma,y_{\sigma,s}}$ (note that $\hat{y}_{\sigma,s}$ is not necessarily in P). We estimate the distance from q to $y_{\sigma,s}$ by $d_X(q,\hat{y}_{\sigma,s})+d_X(\hat{y}_{\sigma,s},y_{\sigma,s})$. We estimate the distance from q to $y_{\sigma,p}$ in a symmetric fashion. Overall we estimated the distance from q to at most 2τ points. Our answer will be the point with smallest estimated distance among those. Clearly the query time is $O(\tau\log\log N)$.

Finally, we argue that the returned point is a 2ρ approximation to the nearest neighbor. Let $x\in P$ be the closest point to q. Suppose w.l.o.g. that in the special ordering $\sigma'\in\Sigma$ it holds that $q\prec_{\sigma'}x$ (the other case is symmetric). For every pair of points $q\prec_{\sigma'}a\prec_{\sigma'}b\prec_{\sigma'}x$, it holds that $d_X(a,b)\le\rho\cdot d_X(q,x)$. Suppose w.l.o.g. that we returned the point $y_{\tilde{\sigma},s}$. It holds that

$$\begin{aligned}d_X(q,y_{\tilde{\sigma},s}) &\le d_X(q,\hat{y}_{\tilde{\sigma},s})+d_X(\hat{y}_{\tilde{\sigma},s},y_{\tilde{\sigma},s})\\ &\le d_X(q,\hat{y}_{\sigma',s})+d_X(q,\hat{y}_{\sigma',s},y_{\sigma',s})\\ &\le\rho\cdot d_X(q,x)+\rho\cdot d_X(q,x)=2\rho\cdot d_X(q,x)\ .\end{aligned}$$

The theorem now follows. □

By plugging in the triangle-LSO's of Theorem 3, or that of [FL22] (Table 1) into the Theorem 8 we conclude:

Corollary 5. *Consider an N-point metric space (X,d_X) of doubling dimension d. Then for every $t\in[\Omega(1),d]$, X admits a dynamic labeled t-NNS with label size $O(2^{O(d/t)}\cdot d\cdot\log^2 t\cdot\log N)$, space $O(n\cdot 2^{O(d/t)}\cdot d\cdot\log^2 t\cdot\log N)$, query time $O(2^{O(d/t)}\cdot d\cdot\log^2 t\cdot\log\log N)$, and deletion/insertion in $O(2^{O(d/t)}\cdot d\cdot\log^2 t\cdot\log\log N)$ amortized time.*

A case of special interest is when the stretch is d. Here we get label size $\tilde{O}(d)\cdot\log N$, space $\tilde{O}(d)\cdot n\cdot\log N$, query time $\tilde{O}(d)\cdot\log\log N$, and deletion/insertion in $\tilde{O}(d)\cdot\log\log N$ amortized time.

Corollary 6. *Consider an N-point metric space (X, d_X). Then for every integer k, and $\epsilon \in (0, \frac{1}{2})$, X admits a dynamic labeled $4k + \epsilon$-NNS with label size $O(N^{\frac{1}{k}} \cdot \log^2 N \cdot \frac{k^2}{\epsilon} \cdot \log \frac{k}{\epsilon})$, space $O(n \cdot \mathbb{N}^{\frac{1}{k}} \cdot \log^2 N \cdot \frac{k^2}{\epsilon} \cdot \log \frac{k}{\epsilon})$, query time $O(N^{\frac{1}{k}} \cdot \log N \log\log N \cdot \frac{k^2}{\epsilon} \cdot \log \frac{k}{\epsilon})$, and deletion/insertion in $O(N^{\frac{1}{k}} \cdot \log N \log\log N \cdot \frac{k^2}{\epsilon} \cdot \log \frac{k}{\epsilon})$ amortized time.*

A case of special interest is where the have stretch $\log N$, label size $\tilde{O}(\log^4 N)$, space $n \cdot \tilde{O}(\log^4 N)$, query time $\tilde{O}(\log^3 N)$, and deletion/insertion in $\tilde{O}(\log^3 N)$ amortized time. We make significant improvements on this in the next subsection.

5.4 General Metrics Spaces

In Corollary 6 we obtained a labeled NNS for general metric spaces. However, for the case of stretch $\approx k$, for $k \ll \log N$, the query time is rather large: $\tilde{O}(N^{\frac{1}{k}})$. In this section we show that by slightly increasing the stretch to $8(1+\epsilon)k$, we can construct labeled NNS with extremely small query time: $O(\frac{1}{\epsilon} \cdot \log\log N)$ (Theorem 12). We then focus on the case of stretch $\log N$. Here Corollary 6 provides label size $\tilde{O}(\log^4 N)$ and query time $\tilde{O}(\log^3 N)$. We show a host of possible trade-offs for this stretch case, starting with label size $O(\log^2 N)$ and query time $O(\log\log N)$ (Theorem 12), and culminating with label size $O(1)$ and query time $O(\log\log N)$ (Corollary 9), under the assumption that the aspect ratio is polynomial, and the label size is bounded only in expectation. See Table 7 for a summary of results.

Our approach here is a reduction of the general case to that of ultrametrics (in similar manner to the distance labeling of Mendel Naor [MN07]), using either the classic Ramsey trees [MN07], or the novel clan embeddings [FL21]. We also look into the improved guarantees that we can obtain using the following relaxations: (1) allowing the data structure to fail with some small probability, (2) giving an upper bound on the label size only in expectation (3) assuming that the metric space has polynomial aspect ratio. In Section 5.4.1 we solve the problem for the important special case of ultrametrics. In Sections 5.4.2 and 5.4.3 we reduce from general space to ultrametrics using Ramsey trees, and clan embedding respectively. See Table 7 for a summary of our results.

5.4.1 Ultrametric

A key component for the NNS of general metric space is the case of ultrametrics. We refer to Section 4 for the introduction and definition of ultrametric. Let π be a pre order of the leaves in U. For a vertex x, let π_x be the position (index) of x in π. We will treat π an an order $\preceq_\pi$. Let $P \subseteq U$ be a subset of the points of size n. We construct a Y-fast-trie of the vertices w.r.t. the indices π_x. Note that the space required by the Y-fast-trie is $O(n)$, while in $O(\log\log N)$ time we can query predecessor/successor of a point, and add/remove a point.

Consider a query point $q \in U$, we use the Y-fast-trie (in $O(\log\log N)$ time) to find the predecessor p, and successor s of q w.r.t. π. For every vertex x such that $q \preceq_\pi s \preceq_\pi x$, let y be the lca of q and x, it holds that s is also a descendant of y. Thus, $d_U(q, s) \le d_U(q, x)$. The same argument holds for p, and vertices x such that $x \preceq_\pi p \preceq_\pi q$. We conclude that $\min\{d_U(q, s), d_U(q, p)\} = \min_{x \in P}\{d_U(q, x)\}$.[10]

Until now we only used $O(n)$ space, and the labels are the indices π_x (a single word). To solve the labeled NNS problem, it remains to add some information to help us distinguish between $d_U(q, s)$ and $d_U(q, p)$. We present three different solutions to the problem of estimating distances, under

[10] Alternative formulation of this fact due to [FL22] is that the preorder of an ultrametric is a $(1, 1)$-triangle LSO.

Stretch	Label	Update	Query	Ref	Notes
$8(1+\epsilon)k$	$O(\frac{k}{\epsilon}N^{\frac{1}{k}}\cdot\log N)$	$O(\frac{k}{\epsilon}N^{\frac{1}{k}}\cdot\log\log N)$	$O(\frac{1}{\epsilon}\cdot\log\log N)$	Thm. 12	
$16k$	$O(N^{\frac{1}{k}}\cdot\log N)$	$O(N^{\frac{1}{k}}\cdot\log\log N)$	$O(\log\log N)$	Thm. 14	$\mathbb{E}$
$8(1+\epsilon)k$	$O(\frac{k}{\epsilon}N^{\frac{1}{k}}\cdot\log\frac{1}{\epsilon})$	$O(\frac{k}{\epsilon}N^{\frac{1}{k}}\cdot\log\log N)$	$O(\frac{1}{\epsilon}\cdot\log\log N)$	Cor. 7	Φ
$16(1+\epsilon)k$	$O(N^{\frac{1}{k}}\cdot\log\frac{1}{\epsilon})$	$O(N^{\frac{1}{k}}\cdot\log\log N)$	$O(\log\log N)$	Cor. 9	$\mathbb{E},\Phi$
$8(1+\epsilon)k$	$O(\frac{k}{\epsilon}N^{\frac{1}{k}}\cdot\frac{\log\frac{1}{\delta}}{\epsilon^2})$	$O(\frac{k}{\epsilon}N^{\frac{1}{k}}\cdot\log\log N)$	$\frac{1}{\epsilon}\cdot O(\log\log N+\frac{\log\frac{1}{\delta}}{\epsilon^2})$	Cor. 8	δ
$16(1+\epsilon)k$	$O(N^{\frac{1}{k}}\cdot\frac{\log\frac{1}{\delta}}{\epsilon^2})$	$O(N^{\frac{1}{k}}\cdot\log\log N)$	$O(\log\log N+\frac{\log\frac{1}{\delta}}{\epsilon^2})$	Cor. 10	$\mathbb{E},\delta$
$O(\log N)$	$O(\log^2 N)$	$\tilde{O}(\log N)$	$O(\log\log N)$	Thm. 12	
$O(\log N)$	$O(\log N)$	$O(\log\log N)$	$O(\log\log N)$	Thm. 14	$\mathbb{E}$
$O(\log N)$	$O(\log N)$	$\tilde{O}(\log N)$	$O(\log\log N)$	Cor. 7	Φ
$O(\log N)$	$O(1)$	$\tilde{O}(\log N)$	$O(\log\log N)$	Cor. 9	$\mathbb{E},\Phi$
$O(\log N)$	$O(\log N\cdot\log\frac{1}{\delta})$	$\tilde{O}(\log N)$	$O(\log\log N+\log\frac{1}{\delta})$	Cor. 8	δ
$O(\log N)$	$O(\log\frac{1}{\delta})$	$O(\log\log N)$	$O(\log\log N+\log\frac{1}{\delta})$	Cor. 10	$\mathbb{E},\delta$

Table 7: Different labeled NNS data structures for general metrics and their parameters. We also illustrate all the results for the regime of stretch $O(\log N)$. Here $k\in\mathbb{N}$, and $\epsilon,\delta\in(0,1)$. The space in all the cases above equals n times the label size. $\mathbb{E}$ denotes that the labels size, space, and update time guarantees are only in expectation (the query time is always worst case). Φ denotes that the result holds only for metric spaces with polynomial aspect ratio. δ denotes that the data structure can return a wrong answer with probability δ.

different assumptions. Each yielding a labeled NNS for ultrametrics. Our results are summarized in Table 8.

5.4.1.1 Least common ancestor labeling - exact Our first approach is to retrieve the exact distances $d_U(q,s),d_U(q,p)$. Given a rooted tree, least common ancestor (LCA) labeling is a scheme where each vertex v is assigned a label ℓ_v, and there is an algorithm that given two labels ℓ_v,ℓ_u, can determine the LCA of v and u. We will be using the following scheme by Peleg [Pel05]. The query time is implicit using binary search.

Theorem 9 ([Pel05])**.** *Given an N vertex tree T where each vertex has unique identifier (that can be stored as machine word), there is an algorithm that assign $O(\log N)$ (words) label to each vertex, such that given two labels, in $O(\log\log N)$ time the algorithm return the unique identifier of their lca.*

When there are no unique identifiers, and one only required to return the assigned label, much sparser schemes exist (see [AGKR04, AHL14]). We obtained a labeled-NNS for ultrametric.

Lemma 4. *For every ultrametric U, one can construct exact labeled 1-NNS with label size $O(\log N)$, space $O(n\cdot\log N)$, and query and insertion/deletion time $O(\log\log N)$. Furthermore, the data structure also outputs the exact distance to the closest neighbor.*

Proof. Consider an ultrametric U, where each vertex x has a label $\ell_U(x)$. The unique identifier of an internal node x will be $(x,\ell_U(x))$, while the unique identifier of a leaf vertex x will be (x,π_x).

Stretch	Label	Update	Query	Ref	Notes
1	$O(\log N)$	$O(\log\log N)$	$O(\log\log N)$	Lemma 4	
$1+\epsilon$	$O(\log\frac{1}{\epsilon})$	$O(\log\log N)$	$O(\log\log N)$	Lemma 5	Polynomial aspect ratio
$1+\epsilon$	$O(\frac{\log\frac{1}{\delta}}{\epsilon^2})$	$O(\log\log N)$	$O(\log\log N+\frac{\log\frac{1}{\delta}}{\epsilon^2})$	Lemma 6	Failure probability δ

Table 8: Different labeled NNS data structures for ultrametrics and their parameters. The space in all the cases above equals n times the label size.

Next, we use Theorem 9 to construct lca-labeling scheme $\{L_U(x)\}_{x\in U}$. Finally, in our labeled NNS scheme, the label of each vertex x will be a union of the unique identifier, and the lca-label.

Given a subset P, in addition to the Y-fast-trie above, we will also store all the labels. Given a query point $q\in U$, we will proceed as above. Given the predecessor p, and successor s, we will simply compute $d_U(q,s), d_U(q,p)$ (as they are the stored labels of the lca's), and return the minimum (together with its distance).

The query time consist of querying the Y-fast-trie, and the lca data structures, taking $O(\log\log N)$ in total. The space is simply storing the labels and identifiers, which is $O(\log N)$ words per vertex and $O(n\cdot\log N)$ overall. One can add and remove vertices from P by simply updating the Y-fast-trie, and the list of vertices in P, which is $O(\log\log N)$ per operation. □

5.4.1.2 Distance labeling in trees - $1+\epsilon$ approximation Given a metric space (X,d), *distance labeling* is an algorithm that assigns to each point $x\in X$ a label $\ell(x)$, such that there is an algorithm $\mathcal{A}$ (oblivious to (X,d)) that provided labels $\ell(x),\ell(y)$ of arbitrary $x,y\in X$, can compute $d_X(u,v)$. Specifically, a distance labeling is said to have *stretch* $t\geq 1$ if

$$\forall x,y\in X,\qquad d(x,y)\leq\mathcal{A}\left(l(x),l(y)\right)\leq t\cdot d(x,y).$$

We refer to [FGK20] for an overview of distance labeling schemes in different regimes (and comparison with metric embedding, see also [Pel00, GPPR04, TZ05, EFN18]). Freedman, Gawrychowski, Nicholson, and Weimann [FGNW17], (improving upon [AGHP16, GKK+01]) showed that for any n-vertex unweighted tree, and $\epsilon\in(0,1)$, one can construct an $(1+\epsilon)$-labeling scheme with labels of $O(\log\frac{1}{\epsilon})$ words.

Theorem 10 ([FGNW17])**.** *For any n-vertex unweighted tree $T=(V,E)$, and parameter $\epsilon\in(0,1)$, there is a distance labeling scheme with stretch $1+\epsilon$, $O(\log\frac{1}{\epsilon})$ label size, and constant query time.*

Given a tree metric T of polynomial aspect ratio, by subdividing the edges one can show that there is an unweighted tree T' with poly$(\frac{n}{\epsilon})$ vertices, and a mapping $f:T\to T'$ with a constant c such that $\forall x,y\in T$, $d_T(x,y)\leq c\cdot d_{T'}(f(x),f(y))\leq(1+\epsilon)\cdot d_T(x,y)$. It follows that (assuming $\epsilon\geq n^{-O(1)}$) Theorem 10 holds also for trees with polynomial aspect ratio.

Lemma 5. *For every ultrametric U with polynomial aspect ratio and $N^{-O(1)}<\epsilon<1$, one can construct labeled $(1+\epsilon)$-NNS with label size $O(\log\frac{1}{\epsilon})$, space $O(n\cdot\log\frac{1}{\epsilon})$, and query and insertion/deletion time $O(\log\log N)$.*

Proof. Consider an ultrametric U with polynomial aspect ratio. We treat it as a tree and use Theorem 10 to obtain a $1+\epsilon$ distance labeling. In our labeled NNS scheme, the label of each vertex x will contain (in addition to previous information) the distance label.

Given a subset P, in addition to the Y-fast-trie above, we will also store all the labels. Given a query point $q \in U$, we will proceed as above. Given the predecessor p, and successor s, we will simply estimate $d_U(q,s), d_U(q,p)$ using the distance label, and return the minimum. Clearly, we obtained a $(1+\epsilon)$-approximate nearest neighbor.

The query time consist of querying the Y-fast-trie, and the distance labeling, taking $O(\log\log N)$ in total. The space usage consist of storing the labels and identifiers, which is $O(\log\frac{1}{\epsilon})$ words per vertex and $O(n \cdot \log\frac{1}{\epsilon})$ overall. One can add and remove vertices from P by simply updating the Y-fast-trie, and the list of vertices in P, which is $O(\log\log N)$ per operation. □

5.4.1.3 Embedding into ℓ_2 and dimension reduction - success with high probability

Here we deal with the case of unbounded aspect ratio, to cope with this situation, we will allow for failure. It is well known fact that every ultrametric embeds isometrically into ℓ_2 (see e.g. [FKM+14]). Specifically, there is a function $f : U \to \mathbb{R}^d$ (for some $d > 0$) such that for every $x, y \in U$, $\|f(x) - f(y)\|_2 = d_U(x,y)$. However, the dimension d could be very large. To reduce the dimension, we will use the Johnson Lindenstrauss dimension reduction [JL84, DG03]. The JL transform is a random linear map of the $\mathbb{R}^d$ into $\mathbb{R}^{O(\frac{\log\frac{1}{\delta}}{\epsilon^2})}$ such that for every $\vec{x}, \vec{y} \in \mathbb{R}^d$ it holds that

$$\Pr[(1-\epsilon)\|\vec{x}-\vec{y}\|_2 \le \|g(\vec{x}) - g(\vec{y})\|_2 \le (1+\epsilon)\|\vec{x}-\vec{y}\|_2] \ge 1-\delta \ .$$

Lemma 6. *For every ultrametric U and parameters $\epsilon, \delta \in [0,1]$, one can construct $(1+\epsilon)$-labeled-NNS with error probability δ. Specifically, given a query q the data structure will return a $(1+\epsilon)$-approximate near neighbor with probability $1-\delta$. The label size is $O(\frac{\log\frac{1}{\delta}}{\epsilon^2})$, the space is $O(n \cdot \frac{\log\frac{1}{\delta}}{\epsilon^2})$, query time $O(\log\log N + \frac{\log\frac{1}{\delta}}{\epsilon^2})$, and insertion/deletion time $O(\log\log N)$.*

Proof. Consider an ultrametric U, let f be the isometric embedding of U into $\mathbb{R}^d$, and g be a random JL transform into $\mathbb{R}^{O(\frac{\log\frac{1}{\delta}}{\epsilon^2})}$. In our labeled NNS scheme, the label of each vertex x will contain (in addition to previous information) the point $g(f(x))$.

Given a subset P, in addition to the Y-fast-trie above, we will also store all the images. Given a query point $q \in U$, we will proceed as above. Given the predecessor p, and successor s, we will simply compute $\|g(f(q)) - g(f(s))\|_2, \|g(f(q)) - g(f(p))\|_2$, and return s or p according to the achieving the minimum. With probability $1-2\delta$, $(1-\epsilon)\|g(f(q) - g(f(s)\|_2 \le d_U(q,s) \le (1+\epsilon)\|g(f(q) - g(f(s)\|_2$ and $(1-\epsilon)\|g(f(q) - g(f(p)\|_2 \le d_U(q,p) \le (1+\epsilon)\|g(f(q) - g(f(p)\|_2$. If this is indeed the case, than clearly we will obtain a $1 + O(\epsilon)$-approximate solution.

The query time consist of querying the Y-fast-trie, and computing the Euclidean distance: $O(\log\log N + \frac{\log\frac{1}{\delta}}{\epsilon^2})$ time in total. The space is simply storing the images and identifiers, which is $O(\frac{\log\frac{1}{\delta}}{\epsilon^2})$ words per vertex and $O(n \cdot \frac{\log\frac{1}{\delta}}{\epsilon^2})$ overall. One can add and remove vertices from P by simply updating the Y-fast-trie, and the list of vertices in P, which is $O(\log\log N)$ per operation (we count here storing the image in $\mathbb{R}^d$ as a single operation). □

5.4.2 Deterministic labeled NNS via Ramsey trees

The following theorem is implicit from the distance oracle construction in Abraham *et al.* [ACE+20].

Theorem 11 ([ACE⁺20]). *Given an N-point metric space (X, d_X) and parameters $k \in \mathbb{N}$, $\epsilon \in (0,1)$, there is a deterministic algorithm that construct a collection $\{U_i\}_{i\in\mathcal{I}}$ of $|\mathcal{I}| = \frac{k}{\epsilon} \cdot N^{\frac{1}{k}}$ ultrametrics over the points of X, such that:*

1. *For every $x, y \in X$ and $i \in \mathcal{I}$, $d_X(x,y) \le d_{U_i}(x,y)$.*
2. *For every $x \in X$, there is a subset $\textbf{home}(x) \subseteq \mathcal{I}$ of $O(\frac{1}{\epsilon})$ ultrametrics, such that for every $y \in X$, $\min_{i\in\textbf{home}(x)} d_{U_i}(x,y) \le 8(1+\epsilon)k \cdot d_X(x,y)$.*

Using Theorem 11, we can construct an labeled NNS:

Theorem 12. *For general metric space (X, d_X), and parameters $k \in \mathbb{N}$ and $\epsilon \in (0,1)$, there is a labeled $8(1+\epsilon)k$-NNS with $O(\frac{k}{\epsilon}N^{\frac{1}{k}} \cdot \log N)$ label size, space $O(n \cdot \frac{k}{\epsilon}N^{\frac{1}{k}} \cdot \log N)$, insertion/deletion time $O(\frac{k}{\epsilon}N^{\frac{1}{k}} \cdot \log\log N)$, and query time $O(\frac{1}{\epsilon} \cdot \log\log N)$.*

Proof. Using Theorem 11, we obtain the collection of ultrametrics $\{U_i\}_{i\in\mathcal{I}}$. For every ultrametric U_i, we will use Lemma 4 to construct a labeled (exact)-NNS for U_i with labels $\{\ell_i(x)\}_{x\in X}$. In our labeled NNS, the label of a point $x \in X$ will consist of all the labels of the ultrametrics $\{\ell_i(x)\}_{i\in\mathcal{I}}$, together with the names of all the ultrametrics in $\textbf{home}(x)$. Thus the label size will be $O(\frac{k}{\epsilon}N^{\frac{1}{k}} \cdot \log N)$.

Given a subset P of points, our data structure will simply be the union of the data structures of all the ultrametrics. Thus a total of $O(n \cdot \frac{k}{\epsilon}N^{\frac{1}{k}} \cdot \log N)$ space. In particular, the insertion/deletion time will be $O(\frac{k}{\epsilon}N^{\frac{1}{k}} \cdot \log\log N)$ (we count the storage of the lca label as a single operation). Given a query $q \in X$, we will go over all the labeled NNS data structures of all the ultrametrics in $\textbf{home}(q)$, and retrieve the nearest neighbor in each one of them (together with it's distance to q). We will return the retrieved point minimizing the distance to q. The correctness is straightforward. The query consist of just querying $O(\frac{1}{\epsilon})$ data structures, and thus $O(\frac{1}{\epsilon} \cdot \log\log N)$ overall. □

By replacing Lemma 4 in the construction above with either Lemma 5 or Lemma 6, we obtain the following:

Corollary 7. *For general metric space (X, d_X) with polynomial aspect ratio, and parameters $k \in \mathbb{N}$ and $\epsilon \in (0,1)$, there is a labeled $8(1+\epsilon)k$-NNS with $O(\frac{k}{\epsilon}N^{\frac{1}{k}} \cdot \log\frac{1}{\epsilon})$ label size, space $O(n \cdot \frac{k}{\epsilon}N^{\frac{1}{k}} \cdot \log\frac{1}{\epsilon})$, insertion/deletion time $O(\frac{k}{\epsilon}N^{\frac{1}{k}} \cdot \log\log N)$, and query time $O(\frac{1}{\epsilon} \cdot \log\log N)$.*

Corollary 8. *For general metric space (X, d_X), and parameters $k \in \mathbb{N}$ and $\epsilon, \delta \in (0,1)$, there is a labeled $8(1+\epsilon)k$-NNS with $O(\frac{k}{\epsilon}N^{\frac{1}{k}} \cdot \frac{\log\frac{1}{\delta}}{\epsilon^2})$ label size, space $O(n \cdot \frac{k}{\epsilon}N^{\frac{1}{k}} \cdot \frac{\log\frac{1}{\delta}}{\epsilon^2})$, insertion/deletion time $O(\frac{k}{\epsilon}N^{\frac{1}{k}} \cdot \log\log N)$, and query time $O(\frac{1}{\epsilon} \cdot (\frac{\log\frac{1}{\delta}}{\epsilon^2} + \log\log N))$. Each query is answered correctly with probability $1-\delta$ (however there are dependencies).*

5.4.3 Labeled NNS via Clan Embeddings

An interesting case for the labeled NNS problem is for stretch $k = O(\log N)$, as here, the label size becomes logarithmic. Specifically, from Theorem 12 one can obtain a labeled $O(\log N)$-NNS with $O(\log^2 N)$ label size, space $O(n \cdot \log^2 N)$, insertion/deletion time $O(\log N \cdot \log\log N)$, and query time $O(\log\log N)$.

If one allows for the space bounds to be only in expectation, we can do much better. We will use the recent clan embedding of Filtser and Le [FL21].

Theorem 13 ([FL21])**.** *Given an N-point metric space (X, d_X) and parameter $k \in \mathbb{N}$, there is a distribution $\mathcal{D}$ over maps f into ultrametrics U, where each point x is mapped into a non-empty subset $f(x)$ of the points, where the subsets $\{f(x)\}_{x \in X}$ are disjoint. It holds that:*

1. *For every $x, y \in X$, and $(f, U) \in \operatorname{supp}(\mathcal{D})$, $d_X(x, y) \le \min_{x' \in f(x), y' \in f(y)} d_U(x', y')$.*
2. *For every point $x \in X$, and $(f, U) \in \operatorname{supp}(\mathcal{D})$, there is a special vertex $\chi(x) \in f(x)$ called the chief of x. For every $y \in X$, it holds that $\min_{y' \in f(y)} d_U(\chi(x), y') \le 16k \cdot d_X(x, y)$.*
3. *For every $x \in X$, $\mathbb{E}_{(f,U) \sim \mathcal{D}}[|f(x)|] = O(N^{\frac{1}{k}})$.*

Using Theorem 11, we can construct a labeled NNS:

Theorem 14. *For general metric space (X, d_X), and parameters $k \in \mathbb{N}$ and $\epsilon \in (0, 1)$, there is a labeled $16k$-NNS with $O(N^{\frac{1}{k}} \cdot \log N)$ label size, space $O(n \cdot N^{\frac{1}{k}} \cdot \log N)$, insertion/deletion time $O(N^{\frac{1}{k}} \cdot \log \log N)$, and query time $O(\log \log N)$.*
The bounds on label size, space, and insertion/deletion time are only in expectation.

Proof. We sample an ultrametric and a map (f, U) using Theorem 14. Next, we use Lemma 4 to construct a labeled (exact)-NNS for U. In our labeled NNS, the label of a point $x \in X$ will consist of the labels of all the points in $f(x) \subset U$, and the identity of the chief $\chi(x) \in f(x)$. Thus the label size is $O(N^{\frac{1}{k}} \cdot \log N)$ in expectation.

Given a subset P of points, our data structure will simply be the data structure from Lemma 4, for all the points $\cup_{x \in P} f(x)$. Thus a total of $O(n \cdot N^{\frac{1}{k}} \cdot \log N)$ space in expectation. In particular, the insertion/deletion time will be $O(N^{\frac{1}{k}} \cdot \log \log N)$ time in expectation. Given a query $q \in X$, we will retrieve the identity of the chief $\chi(q)$, and find the nearest neighbor y' of $\chi(q)$ in $\cup_{x \in P} f(x)$. We will return the point $y \in P$ such that $y' \in f(y)$. The query time is $O(\log \log N)$ (in worst case). To verify correctness, note it holds that

$$d_X(q, y) \le d_U(\chi(q), y') = \min_{z' \in \cup_{z \in P} f(z)} d_U(\chi(q), z') \le 16k \cdot \min_{z \in P} d_X(q, z) \ .$$

□

By replacing Lemma 4 in the construction above with either Lemma 5 or Lemma 6, we obtain the following:

Corollary 9. *For general metric space (X, d_X) with polynomial aspect ratio, and parameters $k \in \mathbb{N}$ and $\epsilon \in (0, 1)$, there is a labeled $16(1+\epsilon)k$-NNS with $O(N^{\frac{1}{k}} \cdot \log \frac{1}{\epsilon})$ label size, space $O(n \cdot N^{\frac{1}{k}} \cdot \log \frac{1}{\epsilon})$, insertion/deletion time $O(N^{\frac{1}{k}} \cdot \log \log N)$, and query time $O(\log \log N)$.*
The bounds on label size, space, and insertion/deletion time are only in expectation.

Corollary 10. *For general metric space (X, d_X), and parameters $k \in \mathbb{N}$ and $\epsilon, \delta \in (0, 1)$, there is a labeled $16(1+\epsilon)k$-NNS with $O(N^{\frac{1}{k}} \cdot \frac{\log \frac{1}{\delta}}{\epsilon^2})$ label size, space $O(n \cdot N^{\frac{1}{k}} \cdot \frac{\log \frac{1}{\delta}}{\epsilon^2})$, insertion/deletion time $O(N^{\frac{1}{k}} \cdot \log \log N)$, and query time $O(\log \log N + \frac{\log \frac{1}{\delta}}{\epsilon^2})$.*
The bounds on label size, space, and insertion/deletion time are only in expectation.

5.5 Lower Bound for General Metrics

In this section we show that even though labeled NNS suppose to return only an approximate nearest neighbor, and not pairwise distances, the classic information theoretic lower bound due to the girth still holds. In particular, it follows that the label size-stretch tradeoff in Corollary 6 is tight up to the constant 4, which could be improved to 2 (see Remark 3).

The girth of an unweighted graph G is the length of the shortest cycle in G. The Erdős girth conjecture [Erd64] states that for any g and N, there exist an N-vertex graph with girth g and $\Omega(N^{1+\frac{2}{g-2}})$ edges. The conjecture is known to holds for $g = 4, 6, 8, 12$ (see [Ben66, Wen91]). However, the best known provable lower bound for general k is due to Lazebnik *et al.* [LUW95], who constructed graph with girth g and $\Omega(n^{1+\frac{4}{3}\cdot\frac{1}{g-2}})$ edges. The following lower bound is conditioned on the Erdős girth conjecture, however it is straightforward that using [LUW95], one can obtain the unconditional lower bound of $\Omega(n^{\frac{2}{3}\cdot\frac{1}{k}})$, on the label size.

Theorem 15. *Fix a stretch parameter $t < 2k+1$. Assuming the Erdős girth conjecture, for every labeling algorithm for the labeled t-NNS problem and N, there is an N-vertex unweighted graph H, for which the labeling algorithm assigns labels of total length $\Omega(n^{1+\frac{1}{k}})$ (in particular maximal label size is $\Omega(n^{\frac{1}{k}})$, the size count here is in bits and not words).*

Proof. Fix $g = 2k+2$, and let $G = (V, E)$ be the unweighted graph from the Erdős girth conjecture with girth g and $\Omega(N^{1+\frac{2}{g-2}}) = \Omega(N^{1+\frac{1}{k}})$ edges. We construct a maximal disjoint collection of length 2 paths in G: $P_1, P_2, \dots, P_s$. Such collection can be constructed greedily. Note that $G \setminus \cup P_i$ contains only isolated edges and vertices. It follows that the collection contains $s = \Omega(N^{1+\frac{1}{k}})$ paths. Next, we construct a subgraph G' of G, by taking each path $P_i = \{x, y\} \circ \{y, z\}$, and keeping only one of the edges $\{x, y\}$ or $\{y, z\}$ (we keep all the edges out of $\cup_i P_i$). Note that if $\{x, y\} \in G'$, then $d_{G'}(y, x) = 1$ while $d_{G'}(y, z) \geq g-1 = 2k+1$ (as $d_{G'}(x, y) < g-1$ will imply that there is a cycle with strictly less than g edges). Let $\mathcal{G}$ be all the subgraphs of G that can be constructed with such a process. Note that $|\mathcal{G}| = 2^s = 2^{\Omega(N^{1+\frac{1}{k}})}$.

Consider two different graphs $G_1, G_2 \in \mathcal{G}$. Let $\{\ell_v^1\}_{v\in V}$, $\{\ell_v^2\}_{v\in V}$ be t-NNS labels produced by the algorithm for G_1 and G_2, respectively. We argue that $\{\ell_v^1\}_{v\in V} \neq \{\ell_v^2\}_{v\in V}$. As $G_1 \neq G_2$, there is some triangle $P_i = \{x, y\} \circ \{y, z\}$ such that $\{x, y\} \in G_1$, while $\{y, z\} \in G_2$. Consider a t-NNS data structure constructed for the set $P = \{x, z\}$, and queried on y. Then in G_1, as $d_{G_1}(y, z) \geq 2k+1 > t = t \cdot d_{G_1}(y, x)$ the data structure must return x as the approximate nearest neighbor. From the other hand, in G_2, $d_{G_2}(y, x) \geq 2k+1 > t = t \cdot d_{G_2}(y, z)$, thus the the data structure must return z as the approximate nearest neighbor. As the two data structures return different answers, they must be different. The data structures answers based only on the labels. We conclude that $\{\ell_v^1\}_{v\in V} \neq \{\ell_v^2\}_{v\in V}$. It follows that every graph in $\mathcal{G}$ has a unique (w.r.t. $\mathcal{G}$) labelings. As $|\mathcal{G}| = 2^{\Omega(n^{1+\frac{1}{k}})}$, there must be some graph G', who's sum of label lengths is $\Omega(n^{1+\frac{1}{k}})$. In particular, the maximal label size of a vertex in G' is $\Omega(n^{\frac{1}{k}})$. □

Remark 3 (Tightness of Theorem 15)**.** *Note that Theorem 15 is only an information theoretic bound on the size of NNS label, and do not take into account the query and update times. Ignoring this parameters, one can obtain a matching upper bound (up to second order terms) by using Matoušek embedding [Mat96] into ℓ_∞. Matoušek showed that for every N-point metric space there is an embedding f of X into $\tilde{(}n^{\frac{1}{k}})$-dimensional ℓ_∞ such that for every $x, y \in X$,*

$d_X(x,y) \le \|f(x)-f(y)\|_\infty \le (2k-1)\cdot d_X(x,y)$. Indeed take the NNS-label of x to be the vector $f(x)$, and on query q, return the point $x \in P$ minimizing $\|f(x)-f(q)\|_\infty$.

6 Metric Spanners

6.1 Path Reporting Low Hop Spanners

The results of this section are summarized in Table 3. We begin Section 6.1.1 with proving three meta Theorems: 16, 17, and 18. Each type of LSO implies a certain quality of a path reporting 2-hop spanner. As a result, we conclude a host of such spanners for different metric spaces (Corollary 11). However, the resulting path reporting time depends linearly on the number of orderings and could be somewhat large. When then tune manually the data structures to obtain significant improvements to the path reporting time. In Section 6.1.6 we deal with general metrics, by used

6.1.1 Path reporting 2-hop spanners from LSO's

In this subsection we prove three meta theorems, constructing path reporting 2-hop spanners given LSO.

Theorem 16 (Meta theorem: LSO implies 2 hop path reporting spanner)**.** *Suppose that the metric space (X,d_X) admits a (τ,ρ)-LSO, for $\rho \in (0,\frac{1}{2})$. Then X admits a path reporting 2-hop $1+2\rho$-spanner with $O(n\tau\log n)$ edges and $O(\tau)$ query time.*

Proof. Let Σ be a (τ,ρ)-LSO assumed by the theorem. For every ordering $\sigma \in \Sigma$, we form an unweighted path graph P_σ with vertex set X and the order of vertices along the path is σ. We construct a 2-hop 1 spanner H_σ for P_n^σ using Theorem 4. Note that for every edge $\{x,y\} \in H_\sigma$, $w(\{x,y\}) = d_X(x,y)$. For every point $x \in \sigma$, we will store the set $E_{x,\sigma}$ as defined in Theorem 4. Our spanner will be $H = \cup_{\sigma\in\Sigma} H_\sigma$. Clearly the number of edges is $O(n\tau\log n)$. To argue stretch and 2-hop, consider a pair of points x,y, and let $\sigma \in \Sigma$ be the ordering that satisfies the ordering property for x and y. There must be $z \in X$ such that $x \preceq_\sigma z \preceq_\sigma y$ and $\{x,z\} \in E_{x,\sigma} \subseteq H_\sigma$,$\{z,y\} \in E_{y,\sigma} \subseteq H_\sigma$. Suppose w.l.o.g. that $d_X(x,z) \le \rho\cdot d_X(x,y)$. By triangle inequality, $d_X(y,z) \le (1+\rho)\cdot d_X(x,y)$. We conclude that

$$d_H(x,y) \le d_X(x,z) + d_X(z,y) = \rho\cdot d_X(x,y) + (1+\rho)\cdot d_X(x,y) = (1+2\rho)\cdot d_X(x,y)\ .$$

Finally, for the query time, in each of the orderings we have a unique 2-hop path between x and y, we can find each such path in $O(1)$ time and compute its weight (as guaranteed by Theorem 4). We return the path of minimum weight found in $O(\tau)$ -time. □

Using the exact same proof, we can also conclude the following meta theorem:

Theorem 17 (Meta theorem: Triangle LSO implies 2 hop path reporting spanner)**.** *Suppose that the metric space (X,d_X) admits a (τ,ρ)-triangle LSO. Then X admits a path reporting 2-hop 2ρ-spanner with $O(n\tau\log n)$ edges and $O(\tau)$ query time.*

The only small difference in the proof is in the stretch analysis. Given $z \in X$ such that $x \preceq_\sigma z \preceq_\sigma y$ and $\{x,z\},\{z,y\} \in H_\sigma$, we have that $d_X(x,z), d_X(y,z) \le \rho\cdot d_X(x,y)$, and hence by the triangle inequality $d_H(x,y) \le d_X(x,z) + d_X(z,y) = 2\rho\cdot d_X(x,y)$.

Theorem 18 (Meta theorem: rooted LSO implies 2 hop path reporting spanner)**.** *Suppose that the metric space (X, d_X) admits a (τ, ρ)-rooted LSO. Then X admits a path reporting 2-hop ρ-spanner with $O(n\tau)$ edges and $O(\tau)$ query time.*

Proof. Let Σ be a (τ, ρ)-rooted-LSO as assumed by the theorem. For every ordering $\sigma \in \Sigma$, we simply store the first point in the order x_σ, and add an edge from each point in y in σ to x_σ. This finished the construction of the spanner H. Clearly the number of edges is $O(n\tau)$. To argue stretch and 2-hop, consider a pair of points u, v, and let $\sigma \in \Sigma$ be the ordering where both u, v participating, and $d_X(u, x_\sigma) + d_X(x_\sigma, v) \le \rho \cdot d_X(u, v)$. As $\{u, x_\sigma\}, \{x_\sigma, v\}$ both belong to the spanner, we found a 2-hop path of stretch ρ. To answer a path query, we can exhaustively search all the orderings v belongs to in $O(\tau)$ total time. □

Remark 4. *The path reporting data structure in the constructions of Theorems 16, 17 and 18 can all be made distributed. Specifically, where S is the number of edges in the spanner, we can give each point $x \in X$ a label of size $O(\frac{S}{n})$, such that given the 2 labels of points x, y, in the same query time we can find the promised 2-hop path in the spanner.*

Using the various existing LSO's (summarized in Table 1), and the meta Theorems 16, 17 and 18, we conclude:

Corollary 11. *Consider an n point metric space (X, d_X). Then:*

1. *For every integer k and $\epsilon \in (0, \frac{1}{2})$, X admits a path reporting 2-hop $4k + \epsilon$-spanner with $\tilde{O}(n^{1+\frac{1}{k}} \cdot \epsilon^{-1})$ edges and $\tilde{O}(n^{\frac{1}{k}} \cdot \epsilon^{-1})$ query time.*

2. *If X has doubling dimension d, then for $\epsilon \in (0, \frac{1}{2})$ X admits a path reporting 2-hop $(1 + \epsilon)$-spanner with $\epsilon^{-O(d)} \cdot n \log n$ edges and $\epsilon^{-O(d)}$ query time.*

3. *If X has doubling dimension d, then for $t = \Omega(1)$, X admits a path reporting 2-hop t-spanner with $2^{-O(d/t)} \cdot \tilde{O}(n)$ edges and $2^{-O(d/t)} \cdot d \cdot \log^2 t$ query time.*

4. *If X is taken from d dimensional Euclidean space, then for $\epsilon \in (0, \frac{1}{2})$, X admits a path reporting 2-hop $(1+\epsilon)$-spanner with $O_d(\epsilon^{-d}) \log \frac{1}{\epsilon} \cdot n \log n$ edges and $O_d(\epsilon^{-d}) \cdot \log \frac{1}{\epsilon}$ query time.*

5. *If X are taken from d dimensional Euclidean space, then for $t \in [8, \sqrt{d}]$ and $\epsilon \in (0, \frac{1}{2})$, X admits a path reporting 2-hop $(1 + \epsilon)t$-spanner with $\tilde{O}(\frac{d^{1.5}}{\epsilon \cdot t}) \cdot e^{\frac{2d}{t^2} \cdot (1 + \frac{8}{t^2})} \cdot n \log n$ edges and $\tilde{O}(\frac{d^{1.5}}{\epsilon \cdot t}) \cdot e^{\frac{2d}{t^2} \cdot (1 + \frac{8}{t^2})}$ query time.*

6. *If X are taken from d dimensional ℓ_p space for $p \in [1, 2]$, then for $t \ge 10$, X admits a path reporting 2-hop t-spanner with $\tilde{O}(d) \cdot e^{O(\frac{d}{t^p})} \cdot n \log n$ edges and $\tilde{O}(d) \cdot e^{O(\frac{d}{t^p})}$ query time.*

7. *If X are taken from d dimensional ℓ_p space for $p \in [2, \infty]$, then X admits a path reporting 2-hop $2 \cdot d^{1 - \frac{1}{p}}$-spanner with $\tilde{O}(d) \cdot n \log n$ edges and $\tilde{O}(d)$ query time.*

8. *If (X, d_X) is the shortest path metric of a graph with treewidth k (in particular a tree for $k = 1$), then X admits a path reporting 2-hop 1-spanner with $O(nk \log n)$ edges and $O(k \log n)$ query time.*

9. *If (X, d_X) is the shortest path metric of a fixed minor free graph (in particular planar graph), then for $\epsilon \in (0, \frac{1}{2})$, X admits a path reporting 2-hop $(1 + \epsilon)$-spanner with $O(\frac{n}{\epsilon} \log^2 n)$ edges and $O(\epsilon^{-1} \log^2 n)$ query time.*

6.1.2 Euclidean Metrics

Consider the Euclidean space and its $(O_d(\epsilon^{-d})\log\frac{1}{\epsilon},\epsilon)$-LSO [CHJ20]. This LSO is defined for the entire space, regardless of the specific n points at hand. In other words, it is "hard coded". As such, given a pair of points $x, y \in \mathbb{R}^d$, we can find the ordering σ in the collection satisfying them in $O_d(1)$ time - independent of ϵ and n. As a result, in the path reporting 2-hop spanner constructed by Theorem 16, we can efficiently return a path.

We provide here a very brief description of the LSO, and refer to [CHJ20] for a precise description. We will describe here only the ordering of the box $[0,1)^d$ (the rest of the space is similar). Chan *et al.* divide the box $[0,1)^d$ into $O(\epsilon)^{-d}$ boxes with side length ϵ. Next they argue that there are $O(\epsilon)^{-d}$ orderings on this smaller boxes, such that any two small boxes are consecutive it one of them. Assume by induction that we have an LSO with $O(\epsilon)^{-d}$ orderings for each of the small box (they are all actually symmetric). They simply implement these orderings in the global $O(\epsilon)^{-d}$ orders of the boxes. As a result we obtain $O(\epsilon)^{-d}$ orderings. Finally, they take this entire construction, and make it with small $O_d(1)$ shifts of the form $(\frac{i}{O_d(1)},\frac{i}{O_d(1)}\dots,\frac{i}{O_d(1)})$. Chan *et al.* showed that for every pair of points x, y, there is a shift such that x, y contained together in a box of diameter $O_d(1)\cdot\|x-y\|_2$. Now, when we partition this box, both points will belong to small boxes of diameter $O_d(\epsilon)\cdot\|x-y\|_2$. The ordering where these two boxes will be consecutive, will satisfy the pair x, y.

Our main observation about this construction, is first that the right shift could be found efficiently (a simple computation). Next, given the shift we can compute the small boxes containing x, y, and pre-compute the right ordering for them (there are only $O_d(\epsilon^{-2d})$ potential queries, we can store the solutions to all). Once we know what is the satisfying ordering for x, y, we can use the data structure from Theorem 16 to compute a 2-hop path in $O(1)$ time. We conclude:

Corollary 12. *Consider a subset X of n point in d dimensional Euclidean space, for $\epsilon\in(0,\frac{1}{2})$, X admits a path reporting 2-hop $(1+\epsilon)$-spanner with $m = O_d(\epsilon^{-d})\log\frac{1}{\epsilon}\cdot n\log n$ edges,*
space $m + O_d(\epsilon^{-2d})\log\frac{1}{\epsilon}$ and $O_d(1)$ query time.

6.1.3 Fixed Minor free graphs

Our construction closely follows the construction of rooted LSO's from [FL22]. For clarity, we will use the notion of shortest path decomposition (abbreviated SPD) introduced by Abraham *et al.* [AFGN22] (see also [Fil20a]).

Definition 8 (SPD depth). *A graph has an SPD of depth 1 if and only if it is a (weighted) path. A graph G has an SPD of depth $k \geq 2$ if there exists a* shortest path *P, such that deleting P from the graph G results in a graph whose connected components all have SPD of depth at most $k-1$.*

It is known that (see [AFGN22]) that fixed minor-free graphs have SPD of depth $k = O(\log n)$. However, the family of bounded SPDdepth graphs is much richer and contains dense graphs with K_r as a minor, for arbitrarily large r. The rest of the section is devoted to constructing a path reporting 2-hop $1+\epsilon$ spanner for graph with SPDdepth k. The number of edges will be $O(n\cdot k\cdot\epsilon^{-1}\cdot\log n)$, and the query time $O(k\cdot\epsilon^{-1})$. As a corollary, we will obtain the path reporting 2-hop $(1+\epsilon)$-spanner for minor free graphs.

Our construction will rely on the following lemma from [Kle02] (see also [Tho04, BFN19a]).

Lemma 7. *Consider a weighted graph $G = (V, E, w)$ with parameter $\epsilon \in (0, 1)$, and let P be a shortest path in G. Then one can find for each vertex $v \in V$ a set of vertices, called* landmarks*, L_v on P of size $|L_v| = O(\frac{1}{\epsilon})$, such that for every $v, u \in V$, if the shortest u, v path intersects P, then there exists $x \in L_v$ and $y \in L_u$ satisfying $d_G(v, x) + d_P(x, y) + d_G(y, u) \le (1 + \epsilon) \cdot d_G(v, u)$. Furthermore, x and y are consecutive w.r.t. P. That is, there are no additional vertices from $L_u \cup L_v$ on P between x and y.*

We will construct the spanner recursively. Let P be a shortest path of G such that every component of $G \setminus P$ has an SPD of depth $k - 1$. Let $\{L_v\}_{v \in V}$ be the set of landmarks provided by Lemma 7 w.r.t. P. For every vertex $v \in V$, denote $l(v) = |L_v|$. We construct an auxiliary tree T_P that initially contains the shortest path P only. For every vertex $v \in V$, let $L_v = \{x_1, \dots, x_{l(v)}\} \subseteq P$. We will add the vertices $\{v_i\}_{i=1}^{l(v)}$, which are $l(v)$ copies of v, to T_P, and for every i connect v_i to $x_i \in L_v$ with an edge of weight $d_G(v, x_i)$. Note that T_P has $\sum_{v \in V} l(v) = O(\frac{n}{\epsilon})$ vertices. The distances in T_P dominates the distances in the original graph. That is for every v_i, u_j copies of v, u respectively, it holds that $d_{T_P}(v_i, u_j) \ge d_G(v, u)$. Additionally, by Lemma 7, for every pair of vertices u, v, such that the shortest path between them intersects P, it holds that $d_{T_P}(v_i, u_j) \le (1 + \epsilon) \cdot d_G(v, u)$, for some pair of "consecutive" v_i, u_j.[11] For the tree T_P we will construct a path reporting 2-hop 1-spanner H_P using [KLMS22]:

Theorem 19 ([KLMS22])**.** *Consider an n point tree metric (X, d_X), then X admits a path reporting 2-hop 1-spanner with $O(n \log n)$ edges and $O(1)$ query time.*

The number of edges in H_P is thus bounded by $O(\frac{n}{\epsilon} \cdot \log(\frac{n}{\epsilon})) = O(\frac{n}{\epsilon} \cdot \log n)$. [12] Note that in the actual spanner H_P all the copies of each vertex will be identified. However, for the purpose of looking up the path, we will keep the distinctions in the data structure.

Let $\{C_1, \dots, C_s\}$ be the set of connected components of $G \setminus P$. For each i, let $G_i = G[C_i]$ be the graph induced by C_i, and let H_i be a path reporting 2-hop $1 + \epsilon$ spanner for G_i constructed recursively. Our final spanner will be H_P union $\cup_i H_i$. The total number of edges is bounded by $O(n \cdot k \cdot \epsilon^{-1} \cdot \log n)$, as the depth of the recursion is k, and each vertex contributes $O(\epsilon^{-1} \cdot \log n)$ edges per level.

Next we describe our query algorithm. The SPD defines a hierarchical partition of the vertices. This partition can be represented by a tree $\mathcal{T}$, where the leaves are real vertices, and each internal node in $\mathcal{T}$ represents a cluster. Each such cluster C, represents some recursive step, and is associated with a shortest path $P(C)$ in $G[C]$. Consider a query pair u, v. Starting from the root of $\mathcal{T}$, for every internal node C containing both u, v we will do the following: consider the removed path $P(C)$, and let $L_v = \{x_1, \dots, x_{l(v)}\} \subseteq P(C)$ and $L_u = \{y_1, \dots, y_{l(v)}\} \subseteq P(C)$ be the respective landmarks of v and u. We go over the landmarks, and for every pair of consecutive landmarks x_i, y_j, we query the data structure of the spanner $H_{P(C)}$ to find a 2-hop path between v_i and u_j (recall that in the tree $T_{P(C)}$, v_i and u_j are leaves attached to x_i, y_j respectively). Note that the returned path is associated with a 2-hop path in the real spanner H between u and v of weight $d_{T_{P(C)}}(v_i, u_j) = d_G(v, x_i) + d_G(x_i, y_j) + d_G(y_j, u)$. We keep only the shortest observed 2-hop path. Note that there are at most $O(\epsilon^{-1})$ consecutive pairs, and as each query of $H_{P(C)}$ takes $O(1)$, the

[11] Interesting to note, the construction of rooted LSO for graphs with SPD k in [FL22] works by constructing these trees T_P where each vertex belong to k trees (with $O(\frac{1}{\epsilon})$ copies per tree), and then constructing $(O(\log n), 1)$-rooted LSO for each tree as a black box.

[12] We can assume that $\epsilon > \frac{1}{n^2}$, as otherwise we can simply return the clique as a spanner (with 1-hop, stretch 1, and $O(1)$ query time).

total time spent on C is only $O(\epsilon^{-1})$. In total, we go over all clusters C in $\mathcal{T}$ containing both u, v and check $O(\epsilon^{-1})$ pairs for each such cluster. We return the shortest observed 2-hop path. The total query time is $O(k \cdot \epsilon^{-1})$.

Finally, we argue that the returned 2-hop path has weight at most $(1+\epsilon) \cdot d_G(u,v)$. It will be enough to show that a path of such weight was considered. Let $Q_{u,v}$ be the shortest u-v path in G. Let C be the first cluster in $\mathcal{T}$ (i.e. closest to the root) where the removed path $P(C)$ and $Q_{u,v}$ intersect (note that C is not necessarily the last cluster containing u, v, i.e. the lca of u and v in $\mathcal{T}$). As $Q_{u,v} \subseteq C$, it follows that the distance between u and v in the induced graph $G[C]$ is $d_{G[C]}(u,v) \le w(Q_{u,v}) = d_G(u,v)$. By Lemma 7, there are consecutive landmarks $x_i \in L_v, y_j \in L_u$ such that

$$\begin{aligned} d_{T_{P(C)}}(v_i, u_j) &= d_{G[C]}(v, x_i) + d_{G[C]}(x_i, y_j) + d_{G[C]}(y_j, u) \\ &\le (1+\epsilon) \cdot d_{G[C]}(v, u) \\ &= (1+\epsilon) \cdot d_G(v, u) \ . \end{aligned}$$

In particular, we queried the pair (v_i, u_j). As a result, one of the 2-hop paths we checked had weight bounded by $(1+\epsilon) \cdot d_G(v,u)$. The theorem follows:

Theorem 20. *Consider an n vertex graph $G = (V, E, w)$ with SPDdepth k. Then for every $\epsilon \in (0, \frac{1}{2})$, the shortest path metric (V, d_G) admits a path reporting 2-hop $(1+\epsilon)$-spanner with $O(n \cdot k \cdot \epsilon^{-1} \cdot \log n)$ edges and $O(k \cdot \epsilon^{-1})$ query time.*

We conclude:

Corollary 13. *Consider an n point graph G which is a fixed minor free, then for every $\epsilon \in (0, \frac{1}{2})$, the shortest path metric (V, d_G) admits a path reporting 2-hop $(1+\epsilon)$-spanner with $O(n \cdot \epsilon^{-1} \cdot \log^2 n)$ edges and $O(\epsilon^{-1} \cdot \log n)$ query time.*

6.1.4 Planar graphs

The spanner construction for planar graphs will be (almost) the exact same one as for fixed minor free case, using the shortest path decomposition. However, the SPD in planar graphs has additional structural properties that can be exploited to obtain $O(\epsilon^{-1})$ query time (in a similar manner to the fact that the distance oracles for planar graph [Kle02, Tho04] have $O(\epsilon^{-1})$ query time, while the very similar minor free counter part distance oracles [AG06] has $O(\epsilon^{-1} \log n)$ query time). We will go briefly over the details.

The SPD of planar graphs comes from cycle separators [LT79]. Consider an n-vertex planar graph G with a specific drawing on the plane. Fix an arbitrary root vertex r, and let T_r be the shortest path tree rooted in r. One can find a pair of neighboring vertices a, b in G such that the cycle $C_{a,b}$ consisting of the edge $\{a, b\}$ and the two unique paths P_a,P_b from r to a and b in T_r is a separator. Specifically the cycle $C_{a,b}$ partitions the plane into interior and exterior parts, where at most $\frac{2}{3}n$ vertices lay in the interior and the exterior of the cycle. The crux is that all the paths in the SPD are taken from cycle separators created from a single shortest path tree T_r. In particular, **all the removed paths are shortest paths in the original graph, and not only in the induced graphs**. We refer to [Kle02] for more details [13].

[13] See also [CFKL20] for a detailed and careful exposition, alas in [CFKL20] the planar graph contains a vortex.

There is a very small change in the construction of the spanner from Theorem 20: when considering a cluster C and a removed path $P(C)$, we will create the landmarks (and define the weights of the auxiliary edges to the landmarks $\{v_i, x_i\}$ in $T_{P(C)}$) w.r.t. to distance in the original graph G and not the induced graph $G[C]$. This is possible as P is an actual shortest path in G. As a result, in the query algorithm for the pair u, v, instead of checking all the clusters C in $\mathcal{T}$ containing both u, v (as we don't know which is the last cluster containing the entire shortest path $Q_{u,v}$ from u to v), it is enough to check the last two paths constituting the cycle separator $C_{a,b} = P_a \cup P_b$ that separated u from v. Indeed, the shortest path $Q_{u,v}$ from u to v in G necessarily intersects either P_a or P_b. Thus applying Lemma 7 on one of P_a, P_b will be enough, regardless of the intersection of $Q_{u,v}$ with previously deleted paths. We conclude that in the query algorithm it is enough to check at most two clusters (instead of $k = O(\log n)$). These two clusters can be found efficiently in $O(1)$ time in $\mathcal{T}$ using an lca query [HT84]. We conclude:

Corollary 14. *Consider an n point planar graph $G = (V, E, w)$, then for every $\epsilon \in (0, \frac{1}{2})$, then the shortest path metric (V, d_G) admits a path reporting 2-hop $(1+\epsilon)$-spanner with $O(n \cdot \epsilon^{-1} \cdot \log^2 n)$ edges and $O(\epsilon^{-1})$ query time.*

6.1.5 Treewidth graphs

We begin with recalling the definition of treewidth:

Definition 9 (Tree decomposition)**.** *A tree decomposition of $G(V, E)$, denoted by $\mathcal{T}$, is a tree satisfying the following conditions:*

1. *Each node $i \in V(\mathcal{T})$ corresponds to a subset of vertices X_i of V (called bags), such that $\cup_{i \in V(\mathcal{T})} X_i = V$.*
2. *For each edge $uv \in E$, there is a bag X_i containing both u, v.*
3. *For a vertex $v \in V$, all the bags containing v make up a subtree of $\mathcal{T}$.*

The width *of a tree decomposition $\mathcal{T}$ is $\max_{i \in V(\mathcal{T})} |X_i| - 1$ and the treewidth of G, is the minimum width among all possible tree decompositions of G.*

Consider a graph $G = (V, E, w)$ with tree decomposition $\mathcal{T}$ of width k. Similarly to the case of planar and minor free graphs, the rooted LSO for treewidth graph is also constructed using separators. Initially the number of bags is at most n. There is a bag X_1 such that if we remove it, the number of bags in each connected component of $\mathcal{T} \setminus X_1$ will be at most $\frac{n}{2}$. Thus by the removal of at most $k+1$ vertices we obtain that each connected component has a tree decomposition of width k, and at most $\frac{n}{2}$ bags. Continue this process for $\log n$ steps and we deleted every vertex. In particular, an n-vertex graph G with treewidth k has $\mathsf{SPDdepth}$ $O(k \cdot \log n)$, where all the removed shortest path are simply single vertices.

The rooted LSO is constructed w.r.t. this "SPD". That is, initially we delete a bag X_1 and have $|X_1| \le k+1$ orderings w.r.t. distance from each vertex $x \in X_1$ (in the spanner we respectively added edges from all the points to X_1). Then we continue recursively with each connected component in the tree decomposition $\mathcal{T} \setminus X_1$. An important detail is that when we delete a bag X_i at some step, we create ordering for every vertex initially in X_i, even if it was deleted at some previous step.[14] Note that we look on components of the tree decomposition and not actual graph. Note also that each vertex could correspond to only a single connected component of the remaining tree

[14] If a vertex was deleted at a previous step it implies that we already created an ordering for it, and the new ordering is redundant. However, for the sake of simplicity we will create a new one.

decomposition (if it belongs to two components, it necessary belong to a deleted bag, a thus not correspond to anything). This process also defines a laminar partition (exactly like in the previous cases). Given a pair of points x, y, consider the last cluster C in this laminar partition containing both x, y (C can be found in $O(1)$ time in $\mathcal{T}$ using an lca query [HT84]). Denote by X_C the bag deleted from C. As C is the last cluster containing both x, y, every path between x and y in G goes through a vertex (originally) in X_C. In particular $d_G(x, y) = \min_{z \in X_C} d_G(x, z) + d_G(z, y)$. We conclude that in order to answer the distance query x, y, it is enough to check only the orderings created for the cluster C w.r.t. the bag X_C. Thus we check at most $k + 1$ orderings.

Corollary 15. *Consider an n point graph $G = (V, E, w)$ with treewidth k, then the shortest path metric (V, d_G) admits a path reporting 2-hop 1-spanner with $O(n \cdot k \cdot \log n)$ edges and $O(k)$ query time.*

6.1.6 General Metrics

A distance oracle is a data structure that given a distance query x, y, returns an estimate $\text{est}(x, y)$ of the distance between x and y. The distance oracle has stretch t if for every pair x, y it holds that $d_X(x, y) \leq \text{est}(x, y) \leq t \cdot d_X(x, y)$. Thorup and Zwick [TZ05] in a celebrated result, for every integer k, constructed a distance oracle with stretch $2k - 1$, space $O(n^{1+\frac{1}{k}} \cdot k)$ and query time $O(k)$. Later, Chechik [Che15] improved their result and constructed a distance oracle with stretch $2k - 1$, space $O(n^{1+\frac{1}{k}})$, and query time $O(1)$.

The path reporting 2-hop spanner for general graphs constructed in Corollary 11 has sub-optimal stretch, and very large query time. Our first observation is that [TZ05] implicitly defines a 2-hop (metric) spanner with $O(n^{1+\frac{1}{k}} \cdot k)$ edges. In particular, we can use [TZ05] query algorithm to find such a path in $O(k)$ time. Note that this spanner is optimal up to second order terms (assuming the Erdős girth conjecture [Erd64]). See Theorem 21 for details.

In the interesting regime of stretch $O(\log n)$, the resulting 2-hop spanner has query time $O(\log n)$. One might hope to reduce it to a smaller query time. Indeed, Kahalon *et al.* [KLMS22] got $O(1)$ query time using Ramsey trees [BLMN05a, MN07, ACE$^+$20, Fil21]. Here one have a collection of trees $\mathcal{T}$, and each point x has a home tree $T_x \in \mathcal{T}$ which approximates the shortest path tree of x (see also Theorem 11). In particular, to find a 2-hop path from x to y it is enough to look for it in T_x. While this method is indeed very efficient, the down side is that the stretch is inherently larger. The constant in the stretch is important, as it governs the exponent in the number of edges.

We bridge between the two extremes, and improve the $O(k)$ query time resulting from [TZ05] exponentially to $O(\epsilon^{-1} \cdot \log 2k)$, while inquiring additional factor of 2 to the stretch (getting stretch $(4k - 2)(1 + \epsilon)$). Our approach is to use Awerbuch, Peleg [AP90] sparse cover (similar to padded partition cover Definition 5), which is a collection of clusters of radius $(2k - 1)\Delta$, such that every ball of radius Δ is contained in one of the clusters. To find the distance from x to y, we will be looking for a minimal diameter cluster containing them both, but due to the padding property, the number of clusters we will need to check will be very small.

Theorem 21 ([TZ05] based 2 hop path reporting spanner)**.** *Consider an n point metric space and integer parameter $k \geq 1$. Then X admits a path reporting 2-hop $(2k - 1)$-spanner with $O(n^{1+\frac{1}{k}} \cdot k)$ edges and $O(k)$ query time.*

Algorithm 1: [TZ05] query: $\texttt{Dist}(v,u,i)$

```
$w \leftarrow v$
while $w \notin B(u)$ do
    $i \leftarrow i+1$
    $(u,v) \leftarrow (v,u)$
    $w \leftarrow p_i(v)$
return $d_X(w,u) + d_X(w,v)$
```

Proof sketch. We begin with a brief overview of the distance oracle of Thorup and Zwick [TZ05]. Recall that in the distance oracle construction of [TZ05], a sequence of sets $X = A_0 \supseteq A_1 \supseteq \cdots \supseteq A_t = \emptyset$ is sampled randomly, by choosing each element of A_{i-1} to be in A_i with probability $n^{-\frac{1}{k}}$. For each $v \in X$ and $0 \le i \le k-1$, define the i-th pivot $p_i(v)$ as the nearest point to v in A_i, and $B_i(v) = \{w \in A_i \,:\, d(v,w) < d(v, A_{i+1})\}$.[15] The *bunch* of v is defined as $B(v) = \bigcup_{0 \le i \le k-1} B_i(v)$. The distance oracle stores in a hash table, for each $v \in X$, all the distances to points in $B(v)$, and also to the pivot $p_i(v)$ vertices. The expect size of $B(v)$ is $O(k \cdot n^{\frac{1}{k}})$, and thus $O(k \cdot n^{1+\frac{1}{k}})$ overall. We will construct a spanner H that will accompany the distance oracle: for every point x, add edges to all the points y in its bunch, and to it's pivots. The overall number of edges in the spanner is $O(k \cdot n^{\frac{1}{k}})$. [16]

The query algorithm for the distance between u, v in [TZ05] is described in Algorithm 1, and takes $O(k)$ time. Crucially, the algorithm returns $d_X(w,u) + d_X(w,v)$, where w is a pivot, or in the bunch of both u and v. In particular, both edges $\{w,u\}, \{w,v\}$ belong to our spanner H. It follows that we can run Algorithm 1, and return the path consisting of the two return edges. The $2k-1$ stretch is guaranteed by [TZ05]. We refer to [TZ05] for proofs and analysis, the theorem now follows. □

Theorem 22 (Sparse cover based hop path reporting spanner)**.** *Consider an n point metric space with aspect ratio Φ. Then for every integer parameter $k \ge 1$ and $\epsilon \in (0, \frac{1}{2})$, X admits a path reporting 2-hop $(1+\epsilon)(4k-2)$-spanner with $O(n^{1+\frac{1}{k}} \cdot \epsilon^{-1} \cdot k \cdot \log \Phi)$ edges with $O(\frac{\log 2k}{\epsilon})$ query time.*

Proof. We will use the following classic sparse covers due to Awerbuch and Peleg [AP90].

Lemma 8 (Sparse cover, [AP90])**.** *Consider an n-point metric space (X, d_X), and a parameter $\Delta > 0$. Then X admits a collection of clusters $\mathcal{C}$ of X points such that:*

1. *Every point $x \in X$ belongs to at most $O(k \cdot n^{\frac{1}{k}})$ clusters in $\mathcal{C}$.*

2. *For every cluster $C \in \mathcal{C}$, there is some point x_C, called the center of C, such that $C \subseteq B_X(x_c, (2k-1) \cdot \Delta)$. In particular, C has diameter at most $(4k-2)\Delta$.*

3. *For every x there come cluster $C \in \mathcal{C}$ such that $B_X(x, \Delta) \subseteq C$.*

Assume w.l.o.g. that the minimum distance in X is 1 (otherwise scale). In particular the maximum distance is Φ. For every integer $i \in [1, \lceil \log_{1+\epsilon} \Phi \rceil]$, we apply Lemma 8 with parameter

[15] We assume that $d(v, \emptyset) = \infty$ (this is needed as $A_k = \emptyset$).

[16] Even though apriori the guarantee on the size is only in expectation, we can repeat the construction several times until we obtain the desired size.

$\Delta_i = (1+\epsilon)^i$ to obtain a collection of clusters (sparse cover) $\mathcal{C}^{(i)}$. For every cluster $C \in \mathcal{C}^{(i)}$ let x_C be the cluster center. We create a spanner H by adding an edge from every point y to the cluster centers x_C, for all the clusters y belongs to in all the distance scales. Note that we added a total of $O(n \cdot \log_{1+\epsilon}\Phi \cdot n^{\frac{1}{k}} \cdot k) = O(n^{1+\frac{1}{k}} \cdot \epsilon^{-1} \cdot k \cdot \log\Phi)$ edges. In our data structure we will simply store the spanner, all the created clusters with their centers. In addition for every point x and scale i, we will store the name of the cluster C of the i'th scale where $B_X(x,\Delta_i) \subseteq C$. Finally, we will also create a $2k-1$-distance oracle with $O(1)$ query time and $O(n^{1+\frac{1}{k}})$ space [Che15]. The total space is $O(n^{1+\frac{1}{k}} \cdot \epsilon^{-1} \cdot k \cdot \log\Phi)$.

Given a query u,v we first use the distance oracle to get an estimation $\text{est}(u,v)$ of $d_X(u,v)$. Next, we go over the indices $\left\lfloor \log_{1+\epsilon}\frac{\text{est}(u,v)}{2k-1}\right\rfloor, \dots, \lceil \log_{1+\epsilon}\text{est}(u,v)\rceil + 1$. For every index i, let $C^{(i)} \in \mathcal{C}^{(i)}$ be the cluster containing $B_X(u,(1+\epsilon)^i)$. If $C^{(i)}$ also contains v, then $C^{(i)}$ will provide us with the 2-hop path: $(u, x_{C^{(i)}}, v)$. We will return the 2-hop path of minimum weight (among the observed paths). Clearly, the query time is $O(\log_{1+\epsilon} 2k) = O(\frac{\log 2k}{\epsilon})$. For correctness of estimate, let i be the unique index such that $(1+\epsilon)^{i-1} \le d_X(u,v) < (1+\epsilon)^i$. Then $(1+\epsilon)^{i-1} \le \text{est}(u,v) \le (2k-1)\cdot(1+\epsilon)^i$. In particular, $\left\lfloor \log_{1+\epsilon}\frac{\text{est}(u,v)}{2k-1}\right\rfloor \le i \le \lceil \log_{1+\epsilon}\text{est}(u,v)\rceil + 1$, thus i is one of the indices we checked. As $v \in B_X(u,(1+\epsilon)^i) \subseteq C^{(i)}$, one of the 2-hop paths we checked been of weight $d_X(u,x_{C^{(i)}}) + d_X(v,x_{C^{(i)}}) \le 2\cdot(2k-1)\cdot(1+\epsilon)^i \le (1+\epsilon)(4k-2)\cdot d_X(u,v)$. The theorem now follows. □

6.2 Fault tolerant spanners

In this section we present an extremely simple construction of fault tolerant spanners based on LSO's, which obtains the optimal linear dependence on the fault parameter f. We prove meta theorem Theorems 24, 25, 25, from which we derive all our results in Corollary 16. See Table 4 for a summary.

We begin with an f-fault tolerant 1-spanner construction for the path graph. The construction is very similar to the classic Theorem 4.

Theorem 23. *The path graph $P_n = (v_1, v_2, \dots, v_n)$ admits a 2-hop f-fault tolerant 1-spanner H with $O(n \cdot f \cdot \log n)$ edges.*

Proof. For simplicity we will assume that $n = 2^\delta$ for some integer δ. Later we can simply construct the spanner for a power of 2 in $[n, 2n)$, and remove the redundant edges. For simplicity we will also assume that f is even (otherwise construct an $f+1$ fault tolerant spanner). Assume by induction that for every $\delta' < \delta$, a 2-hop f-fault tolerant 1-spanner with $(2^{\delta'}\cdot\delta'+1)\cdot(f+1)$ edges exist for the path graph $P_{n'} = P_{2^{\delta'}}$. The base case with 2 vertices and $\delta' = 1$ clearly holds. Next consider $n = 2^\delta$. Add edges from all the vertices to the set of $f+1$ vertices $v_{2^{\delta-1}-\frac{f}{2}}, v_{2^{\delta-1}-\frac{f}{2}+1}, \dots, v_{2^{\delta-1}}, \dots, v_{2^{\delta-1}+\frac{f}{2}}$. Denote this edges by H_0. Next, we use the induction hypothesis on the graph induced by $(v_1, v_2, \dots, v_{2^{\delta-1}})$, and the graph induced by $(v_{2^{\delta-1}+1}, \dots, v_{2^\delta})$. Denote the returned spanners by H_1 and H_2 respectively. By the induction hypothesis, the total number of edges is bounded by

$$\begin{aligned}(2^\delta - 1)\cdot(f+1) + 2\cdot\left(2^{\delta-1}(\delta-1)+1\right)\cdot(f+1) &= \left(2^\delta - 1 + 2^\delta(\delta-1) + 2\right)\cdot(f+1)\\ &= \left(2^\delta\cdot\delta + 1\right)\cdot(f+1)\end{aligned}$$

Consider a faulty set F of at most f vertices, and let $v_i, v_j \notin F$ where $i < j$. If $i \le 2^{\delta-1} < j$, there there is some index $l \in \left\{2^{\delta-1} - \frac{f}{2}, \dots, 2^{\delta-1}, \dots, 2^{\delta-1} + \frac{f}{2}\right\}$ such that $i \le l \le j$ and $v_l \notin F$. In particular,

$\{v_i, v_l\}, \{v_l, v_j\}$ are edges in H, and thus $H \setminus F$ contains a 2 hop shortest path between v_i and v_j. Otherwise, if either $j \le 2^{\delta-1}$ or $i > 2^{\delta-1}$, then by the induction hypotesis one of the spanners $H_1 \setminus F$ or $H_2 \setminus F$ contains a shortest path between v_i and v_j. □

We now turn to our meta theorems.

Theorem 24 (Meta theorem: LSO to 2 hop fault tolerant spanner)**.** *Suppose that a metric space (X, d_X) admits a (τ, ρ)-LSO for $\rho \in (0, \frac{1}{2})$. Then for every integer $f \ge 1$, X admits a 2-hop f-fault tolerant $1 + 2\rho$-spanner with $O(n \cdot f \cdot \tau \cdot \log n)$ edges.*

Proof. Let Σ be the (τ, ρ)-LSO as assumed by the theorem. For every ordering $\sigma \in \Sigma$, we form an unweighted path graph P_σ with vertex set X and the order of vertices along the path is σ. We construct a 2-hop f-fault tolerant 1-spanner H_σ for P_n^σ using Theorem 23. Our spanner will be $H = \cup_{\sigma \in \Sigma} H_\sigma$ (where each edge will get the weight equal to the actual metric distance). Clearly the number of edges is $O(n \cdot f \cdot \tau \cdot \log n)$. To argue stretch and 2-hop, consider a set F of at most f faulty points, and let $x, y \notin F$. Let $\sigma \in \Sigma$ be the ordering that satisfies the LSO property for x and y. There must be $z \notin F$ such that $x \preceq_\sigma z \preceq_\sigma y$ and $\{x, z\}, \{z, y\} \in H_\sigma$. Suppose w.l.o.g. that $d_X(x, z) \le \rho \cdot d_X(x, y)$. By triangle inequality, $d_X(y, z) \le (1 + \rho) \cdot d_X(x, y)$. We conclude that

$$d_{H \setminus F}(x, y) \le d_X(x, z) + d_X(z, y) = \rho \cdot d_X(x, y) + (1 + \rho) \cdot d_X(x, y) = (1 + 2\rho) \cdot d_X(x, y) \ .$$

□

Using the exact same proof, we can also conclude the following meta theorem:

Theorem 25 (Meta theorem: Triangle LSO to 2 hop fault tolerant spanner)**.** *Suppose that a metric space (X, d_X) admits a (τ, ρ)-triangle LSO for $\rho \ge 1$. Then for every integer $f \ge 1$, X admits a 2-hop f-fault tolerant 2ρ-spanner with $O(n \cdot f \cdot \tau \cdot \log n)$ edges.*

The only small difference in the proof is in the stretch analysis. Given $z \in X$ such that $x \preceq_\sigma z \preceq_\sigma y$ and $\{x, z\}, \{z, y\} \in H_\sigma$, we have that $d_X(x, z), d_X(y, z) \le \rho \cdot d_X(x, y)$, and hence $d_H(x, y) \le d_X(x, z) + d_X(z, y) = 2\rho \cdot d_X(x, y)$.

Next we deal with rooted LSO's. Consider the simple star graph (which actually has $(1, 1)$-rooted LSO). Then every 1-fault tolerant spanner with less $\binom{n}{2}$ edges will have stretch ≥ 2. Indeed, if the edge $\{x, y\}$ is missing, and we deleted the middle vertex, then every remaining path will have stretch at least 2. This is the reason that in the meta theorem bellow, the stretch parameter is 2ρ.

Theorem 26 (Meta theorem: rooted LSO to 2 hop fault tolerant spanner)**.** *Suppose that a metric space (X, d_X) admits a (τ, ρ)-rooted LSO for $\rho \ge 1$. Then for every integer $f \ge 1$, X admits a 2-hop f-fault tolerant 2ρ-spanner with $O(n \cdot f \cdot \tau)$ edges.*

Proof. Let Σ be a (τ, ρ)-rooted-LSO as assumed by the theorem. For every ordering $\sigma \in \Sigma$, we construct a spanner H_σ by adding edges from the first $f + 1$ points in σ, to all the other points. The final spanner H is simply the union $H = \cup_{\sigma \in \Sigma} H_\sigma$. Note that as each vertex participates in at most τ orderings, the number of edges is bounded by $O(n \cdot f \cdot \tau)$.

Consider a set F of at most f faulty points, and let $x, y \notin F$. Let $\sigma \in \Sigma$ be the ordering that satisfies the ordering property for x and y. Let $z \notin F$ be the first not deleted point among the first $f + 1$ points in the order σ (note that only f points belong to F). Recall that for the first point x_σ in σ it holds that $d_X(u, x_\sigma) + d_X(x_\sigma, v) \le \rho \cdot d_X(u, v)$. Furthermore, the points are ordered w.r.t.

distance from x_σ and thus $d_X(z, x_\sigma) \le \min\{d_X(u, x_\sigma), d_X(x_\sigma, v)\}$. As $\{u, z\}, \{z, v\}$ both belong to the spanner, we conclude

$$\begin{aligned} d_{H\smallsetminus F}(x,y) &\le d_X(x,z) + d_X(z,y) \\ &\le d_X(x,x_\sigma) + d_X(x_\sigma,z) + d_X(z,x_\sigma) + d_X(x_\sigma,y) \\ &\le 2\cdot(d_X(x,x_\sigma) + d_X(x_\sigma,y)) \le 2\rho\cdot d_X(x,y)\ . \end{aligned}$$

□

Using the various existing LSO's (summarized in Table 1), and the meta Theorems 24, 25 and 26, we conclude:

Corollary 16. *Consider an n point metric space (X, d_X). Then:*

1. *If X has doubling dimension d, then for $\epsilon \in (0, \frac{1}{2})$ X admits a 2-hop f-fault tolerant $(1+\epsilon)$-spanner with $\epsilon^{-O(d)} \cdot f \cdot n \cdot \log n$ edges.*

2. *If X has doubling dimension d, then for $t = \Omega(1)$, X admits a 2-hop f-fault tolerant t-spanner with $2^{-O(d/t)} \cdot f \cdot \tilde{O}(n)$ edges.*

3. *For every integer k and $\epsilon \in (0, \frac{1}{2})$, X admits a 2-hop f-fault tolerant $4k + \epsilon$-spanner with $\tilde{O}(n^{1+\frac{1}{k}} \cdot f \cdot \epsilon^{-1})$ edges.*

4. *If X is taken from d dimensional Euclidean space, then for $\epsilon \in (0, \frac{1}{2})$, X admits a 2-hop f-fault tolerant $(1+\epsilon)$-spanner with $O_d(\epsilon^{-d}) \cdot \log\frac{1}{\epsilon} \cdot f \cdot n \log n$ edges.*

5. *If X are taken from d dimensional Euclidean space, then for $t \in [8, \sqrt{d}]$ and $\epsilon \in (0, \frac{1}{2})$, X admits a 2-hop f-fault tolerant $(1+\epsilon)t$-spanner with $e^{\frac{2d}{t^2}\cdot(1+\frac{8}{t^2})} \cdot \tilde{O}(\frac{d^{1.5}}{\epsilon\cdot t}) \cdot f \cdot n \cdot \log n$ edges.*

6. *If X are taken from d dimensional ℓ_p space for $p \in [1, 2]$, then for $t \in [10, d^{\frac{1}{p}}]$, X admits a 2-hop f-fault tolerant t-spanner with $e^{O(\frac{d}{t^p})} \cdot \tilde{O}(d) \cdot f \cdot n \cdot \log n$ edges.*

7. *If X are taken from d dimensional ℓ_p space for $p \in [2, \infty]$, then X admits a 2-hop f-fault tolerant $2 \cdot d^{1-\frac{1}{p}}$-spanner with $\tilde{O}(d) \cdot f \cdot n \cdot \log n$ edges.*

8. *If (X, d_X) is the shortest path metric of a graph with treewidth k (in particular a tree for $k = 1$), then X admits a 2-hop f-fault tolerant 2-spanner with $O(n \cdot k \cdot f \cdot \log n)$ edges.*

9. *If (X, d_X) is the shortest path metric of a fixed minor free graph (in particular planar graph), then for $\epsilon \in (0, \frac{1}{2})$, X admits a 2-hop f-fault tolerant $2+\epsilon$-spanner with $O(\frac{n}{\epsilon} \cdot f \cdot \log^2 n)$ edges.*

6.3 Reliable Spanners

We begin with a formal definition of reliable spanner.

Definition 10 (Reliable spanner)**.** *A weighted graph H over point set X is a deterministic ν-reliable t-spanner of a metric space (X, d_X) if d_H dominates*[17] *d_X, and for every set $B \subseteq X$ of points, called an* attack set, *there is a set $B^+ \supseteq B$, called a* faulty extension *of B, such that:*

[17] Metric space (X, d_H) dominates metric space (X, d_X) if $\forall u, v \in X$, $d_X(u, v) \le d_H(u, v)$.

1. $|B^+| \leq (1+\nu)|B|$.
2. *For every* $x, y \notin B^+$, $d_{H[X \setminus B]}(x,y) \leq t \cdot d_X(x,y)$.

An oblivious ν*-reliable* t*-spanner is a distribution* $\mathcal{D}$ *over dominating graphs* H*, such that for every attack set* $B \subseteq X$ *and* $H \in \operatorname{supp}(\mathcal{D})$*, there exist a superset* B^+ *of* B *such that, for every* $x, y \notin B^+$*,* $d_{H[X \setminus B]}(x,y) \leq t \cdot d_X(x,y)$*, and* $\mathbb{E}_{H \sim \mathcal{D}}[|B^+|] \leq (1+\nu)|B|$*. We say that the oblivious spanner* $\mathcal{D}$ *has* m *edges if every graph* $H \in \operatorname{supp}(\mathcal{D})$ *has at most* m *edges.*

The distribution $\mathcal{D}$ in Definition 10 is called an oblivious ν-reliable t-spanner because the adversary is oblivious to the specific spanner produced by the distribution (it may be aware to the distribution itself).

Buchin *et al.* [BHO19, BHO20] in two consecutive papers (implicitly) proved that given a (τ,ϵ)-LSO, one can construct a deterministic/oblivious sparse ν-reliable $(1+O(\epsilon))$-spanner. Later, Filtser and Hung [FL22] proved additional meta theorems creating oblivious reliable spanners from triangle (and left sided) LSO's. We will be using the following:

Theorem 27 ([FL22] Meta theorem: triangle LSO implies reliable spanners)**.** *Suppose that a metric space* (X, d_X) *admits a* (τ,ρ)*-Triangle-LSO. Then for every* $\nu \in (0,1)$*,* X *admits an* oblivious ν*-reliable,* (2ρ)*-spanner with* $n\tau \cdot O\left(\log^2 n + \nu^{-1}\tau \log n \cdot \log\log n\right)$ *edges.*

Plugging our newly constructed LSO's (see Table 1), we conclude:

Corollary 17. *Consider an* n *point metric space* (X, d_X)*. Then:*

1. *If* X *has doubling dimension* d*, then for parameters* $t \in [\Omega(1), d]$*,* $\nu \in (0,1)$*,* X *admits an oblivious* ν*-reliable,* $2t$*-spanner with* $\tilde{O}(n) \cdot \nu^{-1} \cdot 2^{O(d/t)}$ *edges.*
2. *If* X *are taken from* d *dimensional Euclidean space, then for parameters* $t \in [8, \sqrt{d}]$*,* $\delta \in (0, \frac{1}{2})$ *and* $\nu \in (0,1)$*,* X *admits a an oblivious* ν*-reliable,* $(1+\delta) \cdot t$*-spanner with* $\nu^{-1} \cdot e^{\frac{4d}{t^2} \cdot (1+\frac{8}{t^2})} \cdot \tilde{O}(n \cdot \frac{d^3}{\epsilon^2 \cdot t^2})$ *edges.*
3. *If* X *are taken from* d *dimensional* ℓ_p *space for* $p \in [1,2]$*, then for parameters* $t \in [10, d^{\frac{1}{p}}]$ *and* $\nu \in (0,1)$*,* X *admits an oblivious* ν*-reliable,* t*-spanner with* $\nu^{-1} \cdot e^{O(\frac{d}{t^p})} \cdot \tilde{O}(n \cdot d^2)$ *edges.*
4. *If* X *are taken from* d *dimensional* ℓ_p *space for* $p \in [2,\infty]$*, then for every* $\nu \in (0,1)$*,* X *admits an oblivious* ν*-reliable,* $2d^{1-\frac{1}{p}}$*-spanner with* $\nu^{-1} \cdot \tilde{O}\left(n \cdot d^2\right)$ *edges.*

See a summary of all state of the art results on reliable spanner in Table 5.

6.4 Light Spanners

Recently, Le and Solomon [LS21] constructed a general framework for constructing light spanners from spanner oracles (originally introduced by [Le20]). Here we show that spanner oracles can be constructed using LSO's, and thus derive new results on light spanners. See Table 6 for a summary. We begin with the required definitions, and meta-theorems from [LS21].

Definition 11 (Spanner Oracle [LS21]). *Let (X, d_X) be a metric space and let $t > 1$ be a stretch parameter. A t-spanner oracle for X, given a subset of terminals $T \subseteq X$ and a distance parameter $L > 0$, outputs a metric spanner S over X such that for every pair of vertices $x, y \in T, x \neq y$ with $L \leq d_G(x, y) < 2L$:*

$$d_S(x, y) \leq t \cdot d_X(x, y).$$

We denote a t-spanner oracle for X by $\mathcal{O}_{X,t}$, and its output subgraph is denoted by $\mathcal{O}_{X,t}(T, L)$, given two parameters $T \subseteq X$ and $L > 0$.

The weak sparsity of a t-spanner oracle $\mathcal{O}_{X,t}$, denoted by $\mathtt{Ws}_{\mathcal{O}_{X,t}}$ as follows:

$$\mathtt{Ws}_{\mathcal{O}_{X,t}} = \sup_{T \subseteq X, L \in \mathbb{R}^+} \frac{w\left(\mathcal{O}_{X,t}(T, L)\right)}{|T|L}$$

Le and Solomon proved two reduction from spanner oracles to light spanners. The first (Theorem 28) is for stretch $t > 2$, while the second (Theorem 29) is for stretch $1 + \epsilon$.

Theorem 28 ([LS21]). *Let (X, d_X) be a metric space with a t-spanner oracle $\mathcal{O}$ of weak sparsity $\mathtt{Ws}_{\mathcal{O}_{X,t}}$ for $t \geq 2$. Then for any $\epsilon > 0$, there exists a $t(1 + \epsilon)$-spanner H for X with lightness $\mathtt{Ws}_{\mathcal{O}_{X,t}} \cdot \tilde{O}\left(\epsilon^{-1}\right)$.*

Theorem 29 ([LS21]). *Let (X, d_X) be a metric space with a $(1 + \epsilon)$-spanner oracle $\mathcal{O}$ of weak sparsity $\mathtt{Ws}_{\mathcal{O}_{X,1+\epsilon}}$ for any $\epsilon > 0$. Then there exists an $(1 + O(\epsilon))$-spanner S for X with lightness $\mathtt{Ws}_{\mathcal{O}_{X,t}} \cdot \tilde{O}\left(\epsilon^{-1}\right) + \tilde{O}\left(\epsilon^{-2}\right)$.*

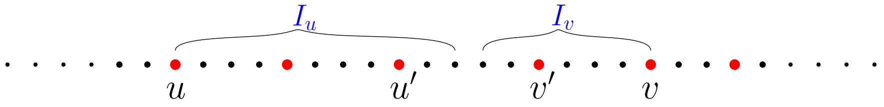
We next show that LSO and triangle LSO imply spanner oracles.

Theorem 30. *Consider a metric space (X, d_X) admitting a (τ, ρ)-LSO for $\rho < \frac{1}{4}$, then X admits a $1 + 8\rho$ spanner oracle $\mathcal{O}_{X,1+8\rho}$ with weak sparsity $\mathtt{Ws}_{\mathcal{O}_{X,1+8\rho}} \leq 2\tau$.*

Proof. Let $T \subseteq X$ be a subset of terminals, and $L > 0$ a parameter. For a pair of terminals $u, v \in T$ and ordering $\sigma \in \Sigma$, we say that u, v are σ-close if $d_X(u, v) \leq 2L$, and there is no terminal $z \in T$ between u and v in σ. Let H be a spanner constructed by adding the edge $\{u, v\}$ for every pair u, v which are σ-close w.r.t. some ordering $\sigma \in \Sigma$.

First we bound the weight of H. Note that for every ordering σ, there are at most $|T| - 1$ σ-close pairs. Further, the weight added to H for each σ-close pair is at most $2L$, as there are only τ orderings, it follows that $w(H) \leq (|T| - 1) \cdot \tau \cdot 2L$. In particular, the weak sparsity of our spanner oracle is bounded by 2τ.

Next we bound the stretch of our spanner by $1 + 8\rho$. Our proof follow similar lines to the spanner construction from LSO in [CHJ20]. The proof is by induction on the distance between two terminals. Consider a pair of terminals $u, v \in T$ such that $d_X(u, v) \leq 2L$, and assume that for every pair of terminals $u', v' \in T$ such that $d_X(u', v') < d_X(u, v)$ it holds that $d_H(u', v') \leq (1 + 8\rho) \cdot d_H(u', v')$. Let $\sigma \in \Sigma$ be an ordering such that (w.l.o.g.) $u \leq_\sigma v$. The points between u and v w.r.t. σ could be partitioned into two consecutive intervals I_u, I_v where $I_u \subseteq B_X(u, \rho \cdot d_X(u, v))$ and $I_v \subseteq B_X(v, \rho \cdot d_X(u, v))$. If u, v are σ-close, then $\{u, v\} \in H$ and we are done. Otherwise, let u' (resp. v') be rightmost (resp. leftmost) terminal in I_u (resp. I_v). Note that it is possible that either $u' = u$ or $v' = v$.



Necessarily u', v' are σ-close, thus $d_H(u',v') = d_X(u',v')$. Furthermore, $d_X(u,u') \le \rho \cdot d_X(u,v)$, and hence by the induction hypothesis, $d_H(u,u') \le (1+8\rho) \cdot d_X(u,u')$. Similarly $d_H(v,v') \le (1+8\rho) \cdot d_X(v,v')$. By triangle inequality $d_X(u',v') \le d_X(u',u) + d_X(u,v) + d_X(v,v') \le (1+2\rho) \cdot d_X(u,v)$. We conclude

$$\begin{aligned} d_H(u,v) &\le d_H(u,u') + d_H(u',v') + d_H(v',v) \\ &\le (1+8\rho) \cdot d_X(u,u') + d_X(u',v') + (1+8\rho) \cdot d_X(v',v) \\ &\le 2\rho(1+8\rho) \cdot d_X(u,v) + (1+2\rho) \cdot d_X(u,v) \\ &\le (1+8\rho) \cdot d_X(u,v) \ . \end{aligned}$$

where in the last inequality we used the assumption $\rho \le \frac{1}{4}$. □

Theorem 31. *Consider a weighted n-vertex graph $G = (V,E,w)$ that admits a (τ,ρ)-triangle-LSO, then G admits:*

- *A 2ρ spanner oracle $\mathcal{O}_{G,2\rho}$ with weak sparsity $\mathtt{Ws}_{\mathcal{O}_{G,2\rho}} \le O(\rho \cdot \tau \cdot \log n)$.*
- *A 3ρ spanner oracle $\mathcal{O}_{G,3\rho}$ with weak sparsity $\mathtt{Ws}_{\mathcal{O}_{G,3\rho}} \le O(\rho \cdot \tau \cdot \log\log n)$.*
- *A 4ρ spanner oracle $\mathcal{O}_{G,4\rho}$ with weak sparsity $\mathtt{Ws}_{\mathcal{O}_{G,4\rho}} \le O(\rho \cdot \tau \cdot \log^* n)$.*

Proof. We begin by proving the first assertion of the theorem. The two others will follows similar lines. Let Σ be the (τ,ρ)-LSO assumed by the theorem. For every ordering $\sigma \in \Sigma$, we form an unweighted path graph $P_{\sigma,T}$ with vertex set T and the order of vertices along the path is σ (delete all the vertices out of T). We construct a 2-hop f-fault tolerant 1-spanner H_σ for $P_{\sigma,T}$ using Theorem 23. Let $H_\sigma^{\le 2\rho L}$ be a spanner of G, where we keep from H_σ only edges of weight at most $\rho \cdot 2L$. Our spanner will be $H = \cup_{\sigma\in\Sigma} H_\sigma^{\le 2\rho L}$.

To bound the weight, note that by Theorem 23, every H_σ consist of $O(|T|\log|T|)$ edges, and thus $H_\sigma^{\le 2\rho L}$ have weight bounded by $O(|T|\log|T| \cdot 2\rho L)$. It follows that the weight of H is bounded by $O(|T|\log|T| \cdot \rho L \cdot \tau)$. We conclude:

$$\mathtt{Ws}_{\mathcal{O}_{G,t}} = \sup_{T\subseteq V, L\in\mathbb{R}^+} \frac{w\left(\mathcal{O}_{G,t}(T,L)\right)}{|T|L} \le \sup_{T\subseteq V, L\in\mathbb{R}^+} \frac{O(|T|\log|T| \cdot \rho L \cdot \tau)}{|T|L} = O(\rho \cdot \tau \cdot \log n) \ . \tag{6}$$

Next we argue that H has stretch at most 2ρ w.r.t. T. Consider a pair of terminals $u, v \in T$ such that $d_X(u,v) \le 2L$. Let $\sigma \in \Sigma$ be an ordering such that (w.l.o.g.) $u \preceq_\sigma v$ and for every a, b such that $u \preceq_\sigma a \preceq_\sigma b \preceq_\sigma v$ it holds that $d_X(a,b) \le \rho \cdot d_X(u,v)$. There is a terminal $x \in T$ such that $u \preceq_\sigma x \preceq_\sigma v$ and $\{u,x\},\{x,v\} \in H_\sigma$. As $d_X(u,x), d_X(v,x) \le \rho \cdot d_X(u,v) \le \rho \cdot 2L$, $\{u,x\},\{x,v\} \in H_\sigma^{\le 2\rho L}$. We conclude

$$d_H(u,v) \le d_{H_\sigma^{\le 2\rho L}}(u,x) + d_{H_\sigma^{\le 2\rho L}}(x,v) = d_X(u,x) + d_X(x,v) \le 2\rho \cdot d_X(u,v) \ ,$$

thus the first assertion of the theorem holds.

Similarly to Theorem 4, one can construct 3-hop 1-spanner for the path with $O(n\log\log n)$ edges, and 4-hop 1-spanner for the path with $O(n\log^* n)$ edges (see [Sol13]). Taking the construction above, and replacing Theorem 4 with these two construction immediately yields the other two assertions. □

Remark 5. *In fact, there is extremely slow growing function $\alpha_h(n)$, such that one can construct h-hop 1-spanner for the path with $O(n \cdot \alpha_h(n))$ edges. We used here $\alpha_2(n) = \lceil \log n \rceil$, $\alpha_3(n) =$*

$\lceil \log\log n \rceil$, and $\alpha_4(n) = \log^* n$. However, there is a general definition, and interestingly for $h = \alpha(n)$, $\alpha_{\alpha(n)}(n) = O(1)$, where $\alpha(n)$ is essentially the inverse Ackermann function. We refer to [Sol13] (see also [KLMS22]) for definitions. As a result, given a (τ,ρ)-triangle-LSO, one can construct a $h\rho$ spanner oracle $\mathcal{O}_{G,h\rho}$ with weak sparsity $\mathtt{Ws}_{\mathcal{O}_{G,h\rho}} \le O(\rho\cdot\tau\cdot\alpha_h(n))$, or a $\alpha(n)\cdot\rho$ spanner oracle $\mathcal{O}_{G,h\rho}$ with weak sparsity $\mathtt{Ws}_{\mathcal{O}_{G,h\rho}} \le O(\rho\cdot\tau)$.

Using the various LSO's (summarized in Table 1), and the meta Theorems 30 and 31 we obtain weak sparsity oracles. Now, plugging in these weak sparsity oracles into the meta Theorems 28 and 29 of [LS21] we conclude:

Corollary 18. *Consider an n point metric space (X, d_X) with doubling dimension d. Then $\forall\epsilon\in(0,\frac{1}{2})$, X admits $(1+\epsilon)$-spanner oracle with weak sparsity $\epsilon^{-O(d)}$. It follows that X admits $(1+\epsilon)$-spanner with lightness $\epsilon^{-O(d)}$.*
Furthermore, for $t=\Omega(1)$, X admits a t-spanner oracle with weak sparsity $2^{O(d/t)}\cdot d\cdot\log^2 t\cdot\log^* n$. It follows that X admits $O(t)$-spanner with lightness $2^{O(d/t)}\cdot d\cdot\log^2 t\cdot\log^* n$.

Using our LSO for d-dimensional Euclidean space, we conclude,

Corollary 19. *For every parameters $t\in[4,2\sqrt{d}]$ and $\epsilon\in(0,1)$, every set of n-point in $\mathbb{R}^d$, admits:*

- *$(1+\epsilon)$-spanner oracle with weak sparsity $O_d(\epsilon^{-d})\cdot\log\frac{1}{\epsilon}$, and thus $(1+\epsilon)$-spanner with lightness $O_d(\epsilon^{-d-1})\cdot\mathrm{polylog}\frac{1}{\epsilon}$.*
- *$(1+\epsilon)2t$-spanner oracle with weak sparsity $e^{\frac{d}{2t^2}\cdot(1+\frac{2}{t^2})}\cdot\tilde{O}(\frac{d^{1.5}}{\epsilon})\cdot\log n$, and thus $(1+\epsilon)4t$-spanner with lightness $e^{\frac{d}{2t^2}\cdot(1+\frac{2}{t^2})}\cdot\tilde{O}(\frac{d^{1.5}}{\epsilon^2})\cdot\log n$.*
- *$(1+\epsilon)3t$-spanner oracle with weak sparsity $e^{\frac{d}{2t^2}\cdot(1+\frac{2}{t^2})}\cdot\tilde{O}(\frac{d^{1.5}}{\epsilon})\cdot\log\log n$, and thus $(1+\epsilon)6t$-spanner with lightness $e^{\frac{d}{2t^2}\cdot(1+\frac{2}{t^2})}\cdot\tilde{O}(\frac{d^{1.5}}{\epsilon^2})\cdot\log\log n$.*
- *$(1+\epsilon)4t$-spanner oracle with weak sparsity $e^{\frac{d}{2t^2}\cdot(1+\frac{2}{t^2})}\cdot\tilde{O}(\frac{d^{1.5}}{\epsilon})\cdot\log^* n$-spanner with lightness $e^{\frac{d}{2t^2}\cdot(1+\frac{2}{t^2})}\cdot\tilde{O}(\frac{d^{1.5}}{\epsilon^2})\cdot\log^* n$.*

Using our LSO for ℓ_p spaces we conclude:

Corollary 20. *For every parameters $t\ge 21$, every set of n-point in d dimensional ℓ_p space for $p\in[1,2]$ admits t-spanner oracle with weak sparsity $e^{O(\frac{d}{t^p})}\cdot\tilde{O}(d\cdot t)\cdot\log^* n$, and thus t-spanner with lightness $e^{O(\frac{d}{t^p})}\cdot\tilde{O}(d\cdot t)\cdot\log^* n$.*

Corollary 21. *Every set of n-point in d dimensional ℓ_p space for $p\in[2,\infty]$ admits $4\cdot d^{1-\frac{1}{p}}$-spanner oracle with weak sparsity $\tilde{O}(d^{2-\frac{1}{p}})\cdot\log^* n$, and thus $5\cdot d^{1-\frac{1}{p}}$-spanner with lightness $\tilde{O}(d^{2-\frac{1}{p}})\cdot\log^* n$.*

References

[ABN11] I. Abraham, Y. Bartal, and O. Neiman. Advances in metric embedding theory. *Advances in Mathematics*, 228(6):3026–3126, 2011, doi:https://doi.org/10.1016/j.aim.2011.08.003. 5

[ABS+20] A. R. Ahmed, G. Bodwin, F. D. Sahneh, K. Hamm, M. J. L. Jebelli, S. G. Kobourov, and R. Spence. Graph spanners: A tutorial review. *Comput. Sci. Rev.*, 37:100253, 2020, doi:10.1016/j.cosrev.2020.100253. 6

[ACE+20] I. Abraham, S. Chechik, M. Elkin, A. Filtser, and O. Neiman. Ramsey spanning trees and their applications. *ACM Trans. Algorithms*, 16(2):19:1–19:21, 2020. preliminary version published in SODA 2018, doi:10.1145/3371039. 12, 32, 33, 42

[ACKW15] I. Abraham, S. Chechik, R. Krauthgamer, and U. Wieder. Approximate nearest neighbor search in metrics of planar graphs. In N. Garg, K. Jansen, A. Rao, and J. D. P. Rolim, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2015, August 24-26, 2015, Princeton, NJ, USA*, volume 40 of *LIPIcs*, pages 20–42. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2015, doi:10.4230/LIPIcs.APPROX-RANDOM.2015.20. 4

[ADD+93] I. Althöfer, G. Das, D. P. Dobkin, D. Joseph, and J. Soares. On sparse spanners of weighted graphs. *Discret. Comput. Geom.*, 9:81–100, 1993, doi:10.1007/BF02189308. 6, 10

[ADF+22] S. Alstrup, S. Dahlgaard, A. Filtser, M. Stöckel, and C. Wulff-Nilsen. Constructing light spanners deterministically in near-linear time. *Theor. Comput. Sci.*, 907:82–112, 2022, doi:10.1016/j.tcs.2022.01.021. 13

[AFGN22] I. Abraham, A. Filtser, A. Gupta, and O. Neiman. Metric embedding via shortest path decompositions. *SIAM J. Comput.*, 51(2):290–314, 2022. a priliminary version apperared in the proceedings of STOC 18, doi:10.1137/19m1296021. 38

[AG06] I. Abraham and C. Gavoille. Object location using path separators. In *Proceedings of the Twenty-fifth Annual ACM Symposium on Principles of Distributed Computing*, PODC '06, pages 188–197, 2006. Full version: https://www.cse.huji.ac.il/~ittaia/papers/AG-TR.pdf, doi:10.1145/1146381.1146411. 40

[AGHP16] S. Alstrup, I. L. Gørtz, E. B. Halvorsen, and E. Porat. Distance labeling schemes for trees. In *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy*, pages 132:1–132:16, 2016, doi:10.4230/LIPIcs.ICALP.2016.132. 31

[AGKR04] S. Alstrup, C. Gavoille, H. Kaplan, and T. Rauhe. Nearest common ancestors: A survey and a new algorithm for a distributed environment. *Theory Comput. Syst.*, 37(3):441–456, 2004, doi:10.1007/s00224-004-1155-5. 30

[AGM12] K. J. Ahn, S. Guha, and A. McGregor. Graph sketches: sparsification, spanners, and subgraphs. In *Proceedings of the 31st ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2012, Scottsdale, AZ, USA, May 20-24, 2012*, pages 5–14, 2012, doi:10.1145/2213556.2213560. 13

[AHL14] S. Alstrup, E. B. Halvorsen, and K. G. Larsen. Near-optimal labeling schemes for nearest common ancestors. In C. Chekuri, editor, *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*, pages 972–982. SIAM, 2014, doi:10.1137/1.9781611973402.72. 30

[AI08] A. Andoni and P. Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. *Commun. ACM*, 51(1):117–122, 2008. Preliminary version published in FOCS 2006, doi:10.1145/1327452.1327494. 10, 15

[AIR18] A. Andoni, P. Indyk, and I. P. Razenshteyn. Approximate nearest neighbor search in high dimensions. *CoRR*, abs/1806.09823, 2018, arXiv:1806.09823. 3, 5, 13

[Als08] B. Alspach. The wonderful Walecki construction. *Bull. Inst. Combin. Appl*, 52:7–20, 2008. see here. 2

[And09] A. Andoni. *Nearest neighbor search: the old, the new, and the impossible.* PhD thesis, Massachusetts Institute of Technology, 2009. see here. 11, 15, 16

[ANN+17] A. Andoni, H. L. Nguyen, A. Nikolov, I. P. Razenshteyn, and E. Waingarten. Approximate near neighbors for general symmetric norms. In H. Hatami, P. McKenzie, and V. King, editors, *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 902–913. ACM, 2017, doi:10.1145/3055399.3055418. 3

[ANRW21] A. Andoni, A. Nikolov, I. P. Razenshteyn, and E. Waingarten. Approximate nearest neighbors beyond space partitions. In D. Marx, editor, *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA 2021, Virtual Conference, January 10 - 13, 2021*, pages 1171–1190. SIAM, 2021, doi:10.1137/1.9781611976465.72. 3

[AP90] B. Awerbuch and D. Peleg. Sparse partitions. In *Proceedings of the 31st IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 503–513, 1990, doi:10.1109/FSCS.1990.89571. 12, 42, 43

[AR15] A. Andoni and I. P. Razenshteyn. Optimal data-dependent hashing for approximate near neighbors. In R. A. Servedio and R. Rubinfeld, editors, *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 793–801. ACM, 2015, doi:10.1145/2746539.2746553. 13

[AS87] N. Alon and B. Schieber. Optimal preprocessing for answering on-line product queries. Technical report, 1987. 8, 14

[ASZ20] A. Andoni, C. Stein, and P. Zhong. Parallel approximate undirected shortest paths via low hop emulators. In K. Makarychev, Y. Makarychev, M. Tulsiani, G. Kamath, and J. Chuzhoy, editors, *Proccedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020, Chicago, IL, USA, June 22-26, 2020*, pages 322–335. ACM, 2020, doi:10.1145/3357713.3384321. 7, 13

[AT11] I. Akgün and B. Ç. Tansel. New formulations of the hop-constrained minimum spanning tree problem via miller-tucker-zemlin constraints. *Eur. J. Oper. Res.*, 212(2):263–276, 2011, doi:10.1016/j.ejor.2011.01.051. 6

[BA92] A. Balakrishnan and K. Altinkemer. Using a hop-constrained model to generate alternative communication network design. *INFORMS J. Comput.*, 4(2):192–205, 1992, doi:10.1287/ijoc.4.2.192. 7

[Bar21] Y. Bartal. Advances in metric ramsey theory and its applications. *CoRR*, abs/2104.03484, 2021, arXiv:2104.03484. 12

[Bas08] S. Baswana. Streaming algorithm for graph spanners - single pass and constant processing time per edge. *Inf. Process. Lett.*, 106(3):110–114, 2008, doi:10.1016/j.ipl.2007.11.001. 13

[BCL02] K. Ball, E. A. Carlen, and E. H. Lieb. *Sharp uniform convexity and smoothness inequalities for trace norms*, pages 171–190. Springer Berlin Heidelberg, Berlin, Heidelberg, 2002, doi:10.1007/978-3-642-55925-9_18. 61

[BDG+21] A. S. Biswas, M. Dory, M. Ghaffari, S. Mitrovic, and Y. Nazari. Massively parallel algorithms for distance approximation and spanners. In K. Agrawal and Y. Azar, editors, *SPAA '21: 33rd ACM Symposium on Parallelism in Algorithms and Architectures, Virtual Event, USA, 6-8 July, 2021*, pages 118–128. ACM, 2021, doi:10.1145/3409964.3461784. 13

[BDMS13] P. Bose, V. Dujmovic, P. Morin, and M. H. M. Smid. Robust geometric spanners. *SIAM J. Comput.*, 42(4):1720–1736, 2013. preliminary version published in SOCG 2013, doi:10.1137/120874473. 9

[BDPW18] G. Bodwin, M. Dinitz, M. Parter, and V. V. Williams. Optimal vertex fault tolerant spanners (for fixed stretch). In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*, pages 1884–1900, 2018, doi:10.1137/1.9781611975031.123. 8

[BDR21] G. Bodwin, M. Dinitz, and C. Robelle. Optimal vertex fault-tolerant spanners in polynomial time. In D. Marx, editor, *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA 2021, Virtual Conference, January 10 - 13, 2021*, pages 2924–2938. SIAM, 2021, doi:10.1137/1.9781611976465.174. 8

[Ben66] C. T. Benson. Minimal regular graphs of girths eight and twelve. *Canadian Journal of Mathematics*, 18:1091–1094, 1966, doi:10.4153/CJM-1966-109-8. 35

[BF18] J. D. Boeck and B. Fortz. Extended formulation for hop constrained distribution network configuration problems. *Eur. J. Oper. Res.*, 265(2):488–502, 2018, doi:10.1016/j.ejor.2017.08.017. 6, 7

[BFH19] A. Bernstein, S. Forster, and M. Henzinger. A deamortization approach for dynamic spanner and dynamic maximal matching. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 1899–1918, 2019, doi:10.1137/1.9781611975482.115. 13

[BFKT22] S. Bhore, A. Filtser, H. Khodabandeh, and C. D. Tóth. Online spanners in metric spaces. In *30th Annual European Symposium on Algorithms, ESA 2022, September 5-7, 2022, Potsdam, Germany*, 2022. to appear, see arXiv version. 13

[BFN19a] Y. Bartal, N. Fandina, and O. Neiman. Covering metric spaces by few trees. In *46th International Colloquium on Automata, Languages, and Programming, ICALP 2019, July 9-12, 2019, Patras, Greece*, pages 20:1–20:16, 2019, doi:10.4230/LIPIcs.ICALP.2019.20. 38

[BFN19b] Y. Bartal, A. Filtser, and O. Neiman. On notions of distortion and an almost minimum spanning tree with constant average distortion. *J. Comput. Syst. Sci.*, 105:116–129, 2019. preliminary version published in SODA 2016, doi:10.1016/j.jcss.2019.04.006. 10

[BHO19] K. Buchin, S. Har-Peled, and D. Oláh. A spanner for the day after. In *35th International Symposium on Computational Geometry, SoCG 2019, June 18-21, 2019, Portland, Oregon, USA*, pages 19:1–19:15, 2019, doi:10.4230/LIPIcs.SoCG.2019.19. 1, 9, 47

[BHO20] K. Buchin, S. Har-Peled, and D. Oláh. Sometimes reliable spanners of almost linear size. In *28th Annual European Symposium on Algorithms, ESA 2020, September 7-9, 2020, Pisa, Italy (Virtual Conference)*, pages 27:1–27:15, 2020, doi:10.4230/LIPIcs.ESA.2020.27. 1, 9, 47

[BKS12] S. Baswana, S. Khurana, and S. Sarkar. Fully dynamic randomized algorithms for graph spanners. *ACM Trans. Algorithms*, 8(4):35:1–35:51, 2012, doi:10.1145/2344422.2344425. 13

[BLMN05a] Y. Bartal, N. Linial, M. Mendel, and A. Naor. On metric Ramsey-type dichotomies. *Journal of the London Mathematical Society*, 71(2):289–303, 2005, doi:10.1112/S0024610704006155. 42

[BLMN05b] Y. Bartal, N. Linial, M. Mendel, and A. Naor. Some low distortion metric ramsey problems. *Discret. Comput. Geom.*, 33(1):27–41, 2005, doi:10.1007/s00454-004-1100-z. 12, 22

[BLW17] G. Borradaile, H. Le, and C. Wulff-Nilsen. Minor-free graphs have light spanners. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science*, FOCS '17, pages 767–778, 2017, doi:10.1109/FOCS.2017.76. 10

[BLW19] G. Borradaile, H. Le, and C. Wulff-Nilsen. Greedy spanners are optimal in doubling metrics. In *Proceedings of the 30th Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '19, pages 2371–2379, 2019, doi:10.1137/1.9781611975482.145. 10

[BP19] G. Bodwin and S. Patel. A trivial yet optimal solution to vertex fault tolerant spanners. In *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing, PODC 2019, Toronto, ON, Canada, July 29 - August 2, 2019*, pages 541–543, 2019, doi:10.1145/3293611.3331588. 8

[BS07] S. Baswana and S. Sen. A simple and linear time randomized algorithm for computing sparse spanners in weighted graphs. *Random Struct. Algorithms*, 30(4):532–563, 2007. preliminary version published in ICALP 2003, doi:10.1002/rsa.20130. 13

[BTS94] H. L. Bodlaender, G. Tel, and N. Santoro. Trade-offs in non-reversing diameter. *Nord. J. Comput.*, 1(1):111–134, 1994. see here. 14

[CCG$^+$98] M. Charikar, C. Chekuri, A. Goel, S. Guha, and S. A. Plotkin. Approximating a finite metric by a small number of tree metrics. In *39th Annual Symposium on Foundations of Computer Science, FOCS '98, November 8-11, 1998, Palo Alto, California, USA*, pages 379–388. IEEE Computer Society, 1998, doi:10.1109/SFCS.1998.743488. 11, 16

[CCGG98] M. Charikar, C. Chekuri, A. Goel, and S. Guha. Rounding via trees: deterministic approximation algorithms for group steiner trees and k-median. In *STOC '98: Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 114–123, New York, NY, USA, 1998. ACM Press, doi:http://doi.acm.org/10.1145/276698.276719. 11

[CDNS95] B. Chandra, G. Das, G. Narasimhan, and J. Soares. New sparseness results on graph spanners. *Int. J. Comput. Geom. Appl.*, 5:125–144, 1995. preliminary version published in SOCG 1992, doi:10.1142/S0218195995000088. 10

[CFKL20] V. Cohen-Addad, A. Filtser, P. N. Klein, and H. Le. On light spanners, low-treewidth embeddings and efficient traversing in minor-free graphs. *CoRR*, abs/2009.05039, 2020. To appear in FOCS 2020,https://arxiv.org/abs/2009.05039, arXiv:2009.05039. 10, 40

[CG06] R. Cole and L. Gottlieb. Searching dynamic point sets in spaces with bounded doubling dimension. In J. M. Kleinberg, editor, *Proceedings of the 38th Annual ACM Symposium on Theory of Computing, Seattle, WA, USA, May 21-23, 2006*, pages 574–583. ACM, 2006, doi:10.1145/1132516.1132599. 13

[CG12] T. H. Chan and A. Gupta. Approximating TSP on metrics with bounded global growth. *SIAM J. Comput.*, 41(3):587–617, 2012. preliminary version published in SODA 2008, doi:10.1137/090749396. 5, 27

[Che15] S. Chechik. Approximate distance oracles with improved bounds. In R. A. Servedio and R. Rubinfeld, editors, *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 1–10. ACM, 2015, doi:10.1145/2746539.2746562. 42, 44

[CHJ20] T. M. Chan, S. Har-Peled, and M. Jones. On locality-sensitive orderings and their applications. *SIAM J. Comput.*, 49(3):583–600, 2020. preliminary version published in ITCS 2019, doi:10.1137/19M1246493. 1, 2, 3, 17, 38, 48

[CLPR10] S. Chechik, M. Langberg, D. Peleg, and L. Roditty. Fault tolerant spanners for general graphs. *SIAM J. Comput.*, 39(7):3403–3423, 2010. preliminary version published in STOC 2009, doi:10.1137/090758039. 8

[Coh00] E. Cohen. Polylog-time and near-linear work approximation scheme for undirected shortest paths. *J. ACM*, 47(1):132–166, 2000. preliminary version published in STOC 1994, doi:10.1145/331605.331610. 7

[CW18] S. Chechik and C. Wulff-Nilsen. Near-optimal light spanners. *ACM Trans. Algorithms*, 14(3):33:1–33:15, 2018. preliminary version published in SODA 2016, doi:10.1145/3199607. 10

[DG03] S. Dasgupta and A. Gupta. An elementary proof of a theorem of johnson and lindenstrauss. *Random Struct. Algorithms*, 22(1):60–65, 2003, doi:10.1002/rsa.10073. 32

[DHN93] G. Das, P. J. Heffernan, and G. Narasimhan. Optimally sparse spanners in 3-dimensional euclidean space. In C. Yap, editor, *Proceedings of the Ninth Annual Symposium on Computational GeometrySan Diego, CA, USA, May 19-21, 1993*, pages 53–62. ACM, 1993, doi:10.1145/160985.160998. 6

[DK11] M. Dinitz and R. Krauthgamer. Fault-tolerant spanners: better and simpler. In *Proceedings of the 30th Annual ACM Symposium on Principles of Distributed Computing, PODC 2011, San Jose, CA, USA, June 6-8, 2011*, pages 169–178, 2011, doi:10.1145/1993806.1993830. 8

[DR20] M. Dinitz and C. Robelle. Efficient and simple algorithms for fault-tolerant spanners. In *PODC '20: ACM Symposium on Principles of Distributed Computing, Virtual Event, Italy, August 3-7, 2020*, pages 493–500, 2020, doi:10.1145/3382734.3405735. 8

[DS17] A. Driemel and F. Silvestri. Locality-Sensitive Hashing of Curves. In *Proceedings of the 33rd International Symposium on Computational Geometry*, volume 77, pages 37:1–37:16, Brisbane, Australia, July 2017. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, doi:10.4230/LIPIcs.SoCG.2017.37. 3

[EFN18] M. Elkin, A. Filtser, and O. Neiman. Prioritized metric structures and embedding. *SIAM J. Comput.*, 47(3):829–858, 2018, doi:10.1137/17M1118749. 31

[EFN20] M. Elkin, A. Filtser, and O. Neiman. Distributed construction of light networks. In *Proceedings of the 39th Symposium on Principles of Distributed Computing*, PODC'20, pages 483—-492, 2020, doi:10.1145/3382734.3405701. 13

[EGK+14] M. Englert, A. Gupta, R. Krauthgamer, H. Räcke, I. Talgam-Cohen, and K. Talwar. Vertex sparsifiers: New results from old techniques. *SIAM J. Comput.*, 43(4):1239–1262, 2014, doi:10.1137/130908440. 23

[Elk11] M. Elkin. Streaming and fully dynamic centralized algorithms for constructing and maintaining sparse spanners. *ACM Trans. Algorithms*, 7(2):20:1–20:17, 2011, doi:10.1145/1921659.1921666. 13

[EN19a] M. Elkin and O. Neiman. Efficient algorithms for constructing very sparse spanners and emulators. *ACM Trans. Algorithms*, 15(1):4:1–4:29, 2019. preliminary version published in SODA 2017, doi:10.1145/3274651. 13

[EN19b] M. Elkin and O. Neiman. Hopsets with constant hopbound, and applications to approximate shortest paths. *SIAM J. Comput.*, 48(4):1436–1480, 2019. preliminary version published in FOCS 2016, doi:10.1137/18M1166791. 13

[EN20] M. Elkin and O. Neiman. Near-additive spanners and near-exact hopsets, A unified view. *Bull. EATCS*, 130, 2020. see here. 13

[ENS15] M. Elkin, O. Neiman, and S. Solomon. Light spanners. *SIAM J. Discret. Math.*, 29(3):1312–1321, 2015, doi:10.1137/140979538. 10, 13

[EP15] M. Elkin and S. Pettie. A linear-size logarithmic stretch path-reporting distance oracle for general graphs. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015*, pages 805–821, 2015, doi:10.1137/1.9781611973730.55. 13

[EP18] I. Z. Emiris and I. Psarros. Products of euclidean metrics and applications to proximity questions among curves. In B. Speckmann and C. D. Tóth, editors, *34th International Symposium on Computational Geometry, SoCG 2018, June 11-14, 2018, Budapest, Hungary*, volume 99 of *LIPIcs*, pages 37:1–37:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018, doi:10.4230/LIPIcs.SoCG.2018.37. 3

[Erd64] P. Erdős. Extremal problems in graph theory. *Theory of Graphs and Its Applications (Proc. Sympos. Smolenice)*, pages 29–36, 1964. see here. 8, 35, 42

[ES16] M. Elkin and S. Solomon. Fast constructions of lightweight spanners for general graphs. *ACM Trans. Algorithms*, 12(3):29:1–29:21, 2016, doi:10.1145/2836167. 13

[FFK20] A. Filtser, O. Filtser, and M. J. Katz. Approximate nearest neighbor for curves - simple, efficient, and deterministic. In A. Czumaj, A. Dawar, and E. Merelli, editors, *47th International Colloquium on Automata, Languages, and Programming, ICALP 2020, July 8-11, 2020, Saarbrücken, Germany (Virtual Conference)*, volume 168 of *LIPIcs*, pages 48:1–48:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020, doi:10.4230/LIPIcs.ICALP.2020.48. 3

[FGK20] A. Filtser, L. Gottlieb, and R. Krauthgamer. Labelings vs. embeddings: On distributed representations of distances. In *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 1063–1075, 2020, doi:10.1137/1.9781611975994.65. 31

[FGNW17] O. Freedman, P. Gawrychowski, P. K. Nicholson, and O. Weimann. Optimal distance labeling schemes for trees. In *Proceedings of the ACM Symposium on Principles of Distributed Computing, PODC 2017, Washington, DC, USA, July 25-27, 2017*, pages 185–194, 2017, doi:10.1145/3087801.3087804. 31

[Fil19] A. Filtser. On strong diameter padded decompositions. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2019, September 20-22, 2019, Massachusetts Institute of Technology, Cambridge, MA, USA*, pages 6:1–6:21, 2019, doi:10.4230/LIPIcs.APPROX-RANDOM.2019.6. 11, 22, 23, 25

[Fil20a] A. Filtser. A face cover perspective to ℓ_1 embeddings of planar graphs. In S. Chawla, editor, *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 1945–1954. SIAM, 2020, doi:10.1137/1.9781611975994.120. 38

[Fil20b] A. Filtser. Scattering and sparse partitions, and their applications. In A. Czumaj, A. Dawar, and E. Merelli, editors, *47th International Colloquium on Automata, Languages, and Programming, ICALP 2020, July 8-11, 2020, Saarbrücken, Germany (Virtual Conference)*, volume 168 of *LIPIcs*, pages 47:1–47:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020, doi:10.4230/LIPIcs.ICALP.2020.47. 23

[Fil21] A. Filtser. Hop-constrained metric embeddings and their applications. In *62nd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2021, Denver, CO, USA, February 7-10, 2022*, pages 492–503. IEEE, 2021, doi:10.1109/FOCS52979.2021.00056. 42

[FKM$^+$14] T. Faver, K. Kochalski, M. K. Murugan, H. Verheggen, E. Wesson, and A. Weston. Roundness properties of ultrametric spaces. *Glasgow Mathematical Journal*, 56(3):519–535, 2014, doi:10.1017/S0017089513000438. 32

[FKN21] A. Filtser, M. Kapralov, and N. Nouri. Graph spanners by sketching in dynamic streams and the simultaneous communication model. In D. Marx, editor, *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA 2021, Virtual Conference, January 10 - 13, 2021*, pages 1894–1913. SIAM, 2021, doi:10.1137/1.9781611976465.113. 13

[FL21] A. Filtser and H. Le. Clan embeddings into trees, and low treewidth graphs. In S. Khuller and V. V. Williams, editors, *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, pages 342–355. ACM, 2021, doi:10.1145/3406325.3451043. 12, 29, 33, 34

[FL22] A. Filtser and H. Le. Locality-sensitive orderings and applications to reliable spanners. In S. Leonardi and A. Gupta, editors, *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20 - 24, 2022*, pages 1066–1079. ACM, 2022, doi:10.1145/3519935.3520042. 1, 2, 3, 9, 10, 11, 22, 23, 27, 28, 29, 38, 39, 47

[FN22] A. Filtser and O. Neiman. Light spanners for high dimensional norms via stochastic decompositions. *Algorithmica*, 2022, doi:10.1007/s00453-022-00994-0. 10

[FS20] A. Filtser and S. Solomon. The greedy spanner is existentially optimal. *SIAM J. Comput.*, 49(2):429–447, 2020. preliminary version published in PODC 2016, doi:10.1137/18M1210678. 10

[GGN06] J. Gao, L. J. Guibas, and A. Nguyen. Deformable spanners and applications. *Computational Geometry*, 35(1):2–19, 2006, doi:https://doi.org/10.1016/j.comgeo.2005.10.001. 6

[GKK⁺01] C. Gavoille, M. Katz, N. A. Katz, C. Paul, and D. Peleg. Approximate distance labeling schemes. In *Algorithms - ESA 2001, 9th Annual European Symposium, Aarhus, Denmark, August 28-31, 2001, Proceedings*, pages 476–487, 2001, doi:10.1007/3-540-44676-1_40. 31

[GKL03] A. Gupta, R. Krauthgamer, and J. R. Lee. Bounded geometries, fractals, and low-distortion embeddings. In *44th Symposium on Foundations of Computer Science (FOCS 2003), 11-14 October 2003, Cambridge, MA, USA, Proceedings*, pages 534–543, 2003, doi:10.1109/SFCS.2003.1238226. 14

[GM03] L. E. N. Gouveia and T. L. Magnanti. Network flow models for designing diameter-constrained minimum-spanning and steiner trees. *Networks*, 41(3):159–173, 2003, doi:10.1002/net.10069. 7

[Got15] L. Gottlieb. A light metric spanner. In *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, pages 759–772, 2015, doi:10.1109/FOCS.2015.52. 10

[GPdSV03] L. E. N. Gouveia, P. Patrício, A. de Sousa, and R. Valadas. MPLS over WDM network design with packet level qos constraints based on ILP models. In *Proceedings IEEE INFOCOM 2003, The 22nd Annual Joint Conference of the IEEE Computer and Communications Societies, San Franciso, CA, USA, March 30 - April 3, 2003*, pages 576–586, 2003, doi:10.1109/INFCOM.2003.1208708. 7

[GPPR04] C. Gavoille, D. Peleg, S. Pérennes, and R. Raz. Distance labeling in graphs. *J. Algorithms*, 53(1):85–112, 2004. preliminary version published in SODA 2001, doi:10.1016/j.jalgor.2004.05.002. 31

[HIM12] S. Har-Peled, P. Indyk, and R. Motwani. Approximate nearest neighbor: Towards removing the curse of dimensionality. *Theory Comput.*, 8(1):321–350, 2012, doi:10.4086/toc.2012.v008a014. 13

[HIS13] S. Har-Peled, P. Indyk, and A. Sidiropoulos. Euclidean spanners in high dimensions. In *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2013, New Orleans, Louisiana, USA, January 6-8, 2013*, pages 804–809, 2013, doi:10.1137/1.9781611973105.57. 10

[HK13] S. Har-Peled and N. Kumar. Approximate nearest neighbor search for low-dimensional queries. *SIAM J. Comput.*, 42(1):138–159, 2013, doi:10.1137/110852711. 13

[HM06] S. Har-Peled and M. Mendel. Fast construction of nets in low-dimensional metrics and their applications. *SIAM J. Comput.*, 35(5):1148–1184, 2006, doi:10.1137/S0097539704446281. 13

[HMO21] S. Har-Peled, M. Mendel, and D. Oláh. Reliable Spanners for Metric Spaces. In *37th International Symposium on Computational Geometry*, SoCG'21, pages 43:1–43:13, 2021. Full version at https://arxiv.org/abs/2007.08738, doi:10.4230/LIPIcs.SoCG.2021.43. 9

[HT84] D. Harel and R. E. Tarjan. Fast algorithms for finding nearest common ancestors. *SIAM J. Comput.*, 13(2):338–355, 1984, doi:10.1137/0213024. 41, 42

[Ind01] P. Indyk. On approximate nearest neighbors under ℓ_∞ norm. *J. Comput. Syst. Sci.*, 63(4):627–638, 2001, doi:10.1006/jcss.2001.1781. 5

[Ind02] P. Indyk. Approximate nearest neighbor algorithms for Fréchet distance via product metrics. In *Proceedings of the 8th Symposium on Computational Geometry*, pages 102–106, Barcelona, Spain, June 2002. ACM Press, doi:10.1145/513400.513414. 3

[IT03] P. Indyk and N. Thaper. Fast image retrieval via embeddings. In *3rd international workshop on statistical and computational theories of vision*, volume 2, page 5. Nice, France, 2003. see here. 3

[JL84] W. Johnson and J. Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. *Contemporary Mathematics*, 26:189–206, 1984. see here. 2, 13, 32

[JLN+05] L. Jia, G. Lin, G. Noubir, R. Rajaraman, and R. Sundaram. Universal approximations for tsp, steiner tree, and set cover. In H. N. Gabow and R. Fagin, editors, *Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, May 22-24, 2005*, pages 386–395. ACM, 2005, doi:10.1145/1060590.1060649. 23

[KL04] R. Krauthgamer and J. R. Lee. Navigating nets: simple algorithms for proximity search. In J. I. Munro, editor, *Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2004, New Orleans, Louisiana, USA, January 11-14, 2004*, pages 798–807. SIAM, 2004. see here. 13

[KL05] R. Krauthgamer and J. R. Lee. The black-box complexity of nearest-neighbor search. *Theor. Comput. Sci.*, 348(2-3):262–276, 2005, doi:10.1016/j.tcs.2005.09.017. 4

[Kle02] P. N. Klein. Preprocessing an undirected planar network to enable fast approximate distance queries. In *Proceedings of the Thirteenth Annual ACM-SIAM Symposium on Discrete Algorithms, January 6-8, 2002, San Francisco, CA, USA*, pages 820–827, 2002. see here. 4, 38, 40

[Kle06] P. N. Klein. Subset spanner for planar graphs, with application to subset TSP. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing*, STOC '06, pages 749–756, 2006, doi:http://doi.acm.org/10.1145/1132516.1132620. 10

[Kle08] P. N. Klein. A linear-time approximation scheme for TSP in undirected planar graphs with edge-weights. *SIAM J. Comput.*, 37(6):1926–1952, 2008, doi:10.1137/060649562. 10

[KLMS22] O. Kahalon, H. Le, L. Milenkovic, and S. Solomon. Can't see the forest for the trees: Navigating metric spaces by bounded hop-diameter spanners. In *Proceedings of the ACM Symposium on Principles of Distributed Computing, PODC 2022, Salerno, Italy, July 25-29, 2022*, 2022. to appear, see arXiv version. 6, 7, 8, 12, 39, 42, 50

[KPX08] I. A. Kanj, L. Perkovic, and G. Xia. Computing lightweight spanners locally. In *Distributed Computing, 22nd International Symposium, DISC 2008, Arcachon, France, September 22-24, 2008. Proceedings*, pages 365–378, 2008, doi:10.1007/978-3-540-87779-0_25. 13

[KW14] M. Kapralov and D. P. Woodruff. Spanners and sparsifiers in dynamic streams. In *ACM Symposium on Principles of Distributed Computing, PODC '14, Paris, France, July 15-18, 2014*, pages 272–281, 2014, doi:10.1145/2611462.2611497. 13

[LCM99] L. J. LeBlanc, J. Chifflet, and P. Mahey. Packet routing in telecommunication networks with path and flow restrictions. *INFORMS J. Comput.*, 11(2):188–197, 1999, doi:10.1287/ijoc.11.2.188. 7

[Le20] H. Le. A PTAS for subset TSP in minor-free graphs. In *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 2279–2298, 2020, doi:10.1137/1.9781611975994.140. 10, 47

[LMS22] H. Le, L. Milenkovic, and S. Solomon. Sparse euclidean spanners with tiny diameter: A tight lower bound. In X. Goaoc and M. Kerber, editors, *38th International Symposium on Computational Geometry, SoCG 2022, June 7-10, 2022, Berlin, Germany*, volume 224 of *LIPIcs*, pages 54:1–54:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022, doi:10.4230/LIPIcs.SoCG.2022.54. 8

[LNS02] C. Levcopoulos, G. Narasimhan, and M. H. M. Smid. Improved algorithms for constructing fault-tolerant spanners. *Algorithmica*, 32(1):144–156, 2002. preliminary version published in STOC 1998, doi:10.1007/s00453-001-0075-x. 8

[LP21] Y. Long and S. Pettie. Planar distance oracles with better time-space tradeoffs. In D. Marx, editor, *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA 2021, Virtual Conference, January 10 - 13, 2021*, pages 2517–2537. SIAM, 2021, doi:10.1137/1.9781611976465.149. 4

[LS21] H. Le and S. Solomon. A unified framework of light spanners II: fine-grained optimality. *CoRR*, abs/2111.13748, 2021, arXiv:2111.13748. 10, 47, 48, 50

[LS22] H. Le and S. Solomon. Near-optimal spanners for general graphs in (nearly) linear time. In J. S. Naor and N. Buchbinder, editors, *Proceedings of the 2022 ACM-SIAM Symposium on Discrete Algorithms, SODA 2022, Virtual Conference / Alexandria, VA, USA, January 9 - 12, 2022*, pages 3332–3361. SIAM, 2022, doi:10.1137/1.9781611977073.132. 13

[LT79] R. Lipton and R. Tarjan. A separator theorem for planar graphs. *SIAM Journal on Applied Mathematics*, 36(2):177–189, 1979. see here. 40

[Luk99] T. Lukovszki. New results on fault tolerant geometric spanners. In F. Dehne, J.-R. Sack, A. Gupta, and R. Tamassia, editors, *Algorithms and Data Structures*, pages 193–204, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg, doi:10.1007/3-540-48447-7_20. 8

[LUW95] F. Lazebnik, V. A. Ustimenko, and A. J. Woldar. A new series of dense graphs of high girth. *Bulletin of the American mathematical society*, 32(1):73–79, 1995, doi:10.1090/S0273-0979-1995-00569-0. 35

[LW21] H. Le and C. Wulff-Nilsen. Optimal approximate distance oracle for planar graphs. In *62nd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2021, Denver, CO, USA, February 7-10, 2022*, pages 363–374. IEEE, 2021, doi:10.1109/FOCS52979.2021.00044. 4

[Mat96] J. Matoušek. On the distortion required for embedding finite metric spaces into normed spaces. *Israel Journal of Mathematics*, 93(1):333–344, 1996, doi:10.1007/BF02761110. 5, 35

[MN07] M. Mendel and A. Naor. Ramsey partitions and proximity data structures. *Journal of the European Mathematical Society*, 9(2):253–275, 2007. 12, 29, 42

[Ngu13] H. L. Nguyen. Approximate nearest neighbor search in ℓ_p. *CoRR*, abs/1306.3601, 2013, arXiv:1306.3601. 11, 20

[NS07] G. Narasimhan and M. H. M. Smid. *Geometric spanner networks*. Cambridge University Press, 2007, doi:10.1017/CBO9780511546884. 6, 10

[OR07] R. Ostrovsky and Y. Rabani. Low distortion embeddings for edit distance. *J. ACM*, 54(5):23, 2007, doi:10.1145/1284320.1284322. 3

[Par22] M. Parter. Nearly optimal vertex fault-tolerant spanners in optimal time: sequential, distributed, and parallel. In S. Leonardi and A. Gupta, editors, *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20 - 24, 2022*, pages 1080–1092. ACM, 2022, doi:10.1145/3519935.3520047. 8

[Pel00] D. Peleg. Proximity-preserving labeling schemes. *J. Graph Theory*, 33(3):167–176, 2000. preliminary version published in WG 1999, doi:10.1002/(SICI)1097-0118(200003)33:3<167::AID-JGT7>3.0.CO;2-5. 31

[Pel05] D. Peleg. Informative labeling schemes for graphs. *Theor. Comput. Sci.*, 340(3):577–593, 2005, doi:10.1016/j.tcs.2005.03.015. 30

[PS03] H. Pirkul and S. Soni. New formulations and solution procedures for the hop constrained network design problem. *Eur. J. Oper. Res.*, 148(1):126–140, 2003, doi:10.1016/S0377-2217(02)00366-1. 7

[RAJ12] A. Rossi, A. Aubry, and M. Jacomino. Connectivity-and-hop-constrained design of electricity distribution networks. *Eur. J. Oper. Res.*, 218(1):48–57, 2012, doi:10.1016/j.ejor.2011.10.006. 7

[Sol13] S. Solomon. Sparse euclidean spanners with tiny diameter. *ACM Trans. Algorithms*, 9(3):28:1–28:33, 2013, doi:10.1145/2483699.2483708. 14, 49, 50

[Sol14] S. Solomon. From hierarchical partitions to hierarchical covers: optimal fault-tolerant spanners for doubling metrics. In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 363–372, 2014, doi:10.1145/2591796.2591864. 8

[Tal04] K. Talwar. Bypassing the embedding: algorithms for low dimensional metrics. In *STOC '04: Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, pages 281–290. ACM Press, 2004, doi:http://doi.acm.org/10.1145/1007352.1007399. 27

[Tho04] M. Thorup. Compact oracles for reachability and approximate distances in planar digraphs. *Journal of the ACM*, 51(6):993–1024, 2004, doi:10.1145/1039488.1039493. 4, 38, 40

[TZ05] M. Thorup and U. Zwick. Approximate distance oracles. *J. ACM*, 52(1):1–24, 2005. preliminary version published in STOC 2001, doi:http://doi.acm.org/10.1145/1044731.1044732. 7, 12, 31, 42, 43

[WA88] K. A. Woolston and S. L. Albin. The design of centralized networks with reliability and availability constraints. *Comput. Oper. Res.*, 15(3):207–217, 1988, doi:10.1016/0305-0548(88)90033-0. 7

[Wen91] R. Wenger. Extremal graphs with no c4's, c6's, or c10's. *Journal of Combinatorial Theory, Series B*, 52(1):113 – 116, 1991, doi:https://doi.org/10.1016/0095-8956(91)90097-4. 35

[Wil83] D. E. Willard. Log-logarithmic worst-case range queries are possible in space theta(n). *Inf. Process. Lett.*, 17(2):81–84, 1983, doi:10.1016/0020-0190(83)90075-3. 11, 25

A Missing Proofs from Section 3

This appendix contains the missing proofs from Section 3, restated for convenience.

Claim 1. *Denote by V_d the volume of a unit ball in $\mathbb{R}^d$. The volume of a hyperspherical cup of radius r at distance $\alpha \cdot r$ (for $\alpha \in (\frac{1}{2\sqrt{d}}, \frac{1}{2})$) from the origin is $\Omega(1) \cdot \frac{1}{\alpha \cdot \sqrt{d}} \cdot (1-\alpha^2)^{\frac{d+1}{2}} \cdot V_d \cdot r^d$.*

Proof. By scaling, it is enough to prove the claim for $r = 1$. To calculate the volume of the cap we will simply integrate over the values of the first coordinate, which take values in $[\alpha, 1]$. Once the first coordinate is fixed to be x, the rest of the coordinate constitute a $d - 1$ dimensional ball of radius $\sqrt{1 - x^2}$. We have that

$$\begin{aligned}
\mathrm{Vol}\left(\text{cup at distance } \alpha\right) &= \int_{\alpha}^{1} V_{d-1} \cdot \sqrt{1 - x^2}^{d-1} dx \\
&= V_{d-1} \cdot \int_{0}^{1-\alpha^2} \frac{y^{\frac{d-1}{2}}}{2\sqrt{1-y}} dy \\
&\geq V_{d-1} \cdot \int_{1-(\frac{3}{2}\alpha)^2}^{1-\alpha^2} \frac{y^{\frac{d-1}{2}}}{2 \cdot \frac{3}{2}\alpha} dy \\
&= V_d \cdot \frac{V_{d-1}}{V_d} \cdot \frac{2}{d+1} \cdot \frac{1}{3\alpha} \cdot y^{\frac{d+1}{2}} \Big|_{1-(\frac{3}{2}\alpha)^2}^{1-\alpha^2} \\
&= \Omega(1) \cdot V_d \cdot \frac{1}{\alpha \cdot \sqrt{d}} \cdot (1 - \alpha^2)^{\frac{d+1}{2}} \cdot \left(1 - \left(\frac{1 - (\frac{3}{2}\alpha)^2}{1 - \alpha^2}\right)^{\frac{d+1}{2}}\right) ,
\end{aligned}$$

where in the second equality follows by substitution $y = 1 - x^2$, and the last equality holds by the fact $\frac{V_{d-1}}{V_d} = \Omega(\sqrt{d})$. The claim holds as $\alpha \geq \frac{1}{2\sqrt{d}}$, and thus $1 - \left(\frac{1-(\frac{3}{2}\alpha)^2}{1-\alpha^2}\right)^{\frac{d+1}{2}} = 1 - \left(1 - \frac{\frac{5}{4}\cdot\alpha^2}{1-\alpha^2}\right)^{\frac{d+1}{2}} \geq 1 - \left(1 - \frac{\frac{5}{4}\cdot\frac{1}{4d}}{1-\frac{1}{4d}}\right)^{\frac{d+1}{2}} = \Omega(1)$. □

Lemma 1. *For $p \in [1, 2]$, and every pair of points $x, y \in \mathbb{R}^d$ at distance $\|x - y\|_p = \frac{1}{t} \in [d^{-\frac{1}{p}}, \frac{1}{3}] \cdot w_i$, the probability that x and y belong to the same clusters in $\mathcal{C}_p^{w_i}$ is bounded by*

$$\frac{Vol_d\left(B_p(x, w_i) \cap B_p(y, w_i)\right)}{Vol_d\left(B_p(x, w_i) \cup B_p(y, w_i)\right)} = e^{-\Omega(\frac{d}{t^p})}$$

Proof. The following is a know fact (ℓ_p is p-uniformly smooth, [18] see e.g. [BCL02]):

$$\text{For } p \in [1, 2] \text{ and } x, y \in \ell_p : \qquad \frac{\|x\|_p^p + \|y\|_p^p}{2} \leq \|\frac{x+y}{2}\|_p^p + \|\frac{x-y}{2}\|_p^p \ . \tag{7}$$

By symmetry we can assume that $w_i = 1$.

Claim 2. *For $p \in [1, 2]$, and $\|x - y\|_p \leq \frac{1}{t} < \frac{1}{3}$, it holds that* $B_p\left(\frac{x+y}{2}, \left(1 - \frac{3}{(2t)^p}\right)^{\frac{1}{p}}\right) \subseteq \left(B_p(x, 1) \cap B_p(y, 1)\right) \cup B_p\left(x, \left(1 - \frac{4}{(2t)^p}\right)^{\frac{1}{p}}\right) \cup B_p\left(y, \left(1 - \frac{4}{(2t)^p}\right)^{\frac{1}{p}}\right)$.

Proof. Suppose for contradiction otherwise, then there is point $z \in B_p\left(\frac{x+y}{2}, \left(1 - \frac{3}{(2t)^p}\right)^{\frac{1}{p}}\right) \setminus \left(B_p(x, 1) \cap B_p(y, 1)\right) \cup B_p\left(x, \left(1 - \frac{4}{(2t)^p}\right)^{\frac{1}{p}}\right) \cup B_p\left(y, \left(1 - \frac{4}{(2t)^p}\right)^{\frac{1}{p}}\right)$. W.l.o.g. $\|x - z\|_p \leq \|y - z\|_p$.

[18] For $p \in (1, 2]$, eq. (7) holds for ℓ_p space due to uniform smoothness. From the other hand, one can easily verify that eq. (7) holds for absolute value, and hence also for ℓ_1.

Hence

$$\begin{aligned}\|y-z\|_p &> 1\\ \|x-z\|_p &> \left(1-\frac{4}{(2t)^p}\right)^{\frac{1}{p}}\\ \|\frac{x+y}{2}-z\|_p &\le \left(1-\frac{3}{(2t)^p}\right)^{\frac{1}{p}}\quad ,\end{aligned}$$

applying eq. (7) to $x-z$ and $y-z$ we have

$$\begin{aligned}\|\frac{x-y}{2}\|_p^p &\ge \frac{\|x-z\|_p^p+\|y-z\|_p^p}{2}-\|\frac{x+y}{2}-z\|_p^p\\ &> \frac{1+\left(1-\frac{4}{(2t)^p}\right)}{2}-\left(1-\frac{3}{(2t)^p}\right)=\frac{1}{(2t)^p}\,,\end{aligned}$$

implying that $\|x-y\|_p > \frac{1}{t}$, a contradiction. □

Denote by V_p the volume of the ℓ_p unit ball. Note that the volume of a ball of radius t equals $t^d\cdot V_p$. By Claim 2 it follows that

$$\begin{aligned}\frac{\mathrm{Vol}\left(\mathrm{B_p}\left(\mathrm{x},1\right)\cap\mathrm{B_p}\left(\mathrm{y},1\right)\right)}{\mathrm{V_p}} &\ge \frac{1}{V_p}\cdot\mathrm{Vol}\left(B_p\left(\frac{x+y}{2},\left(1-\frac{3}{(2t)^p}\right)^{\frac{1}{p}}\right)\right)-\frac{2}{V_p}\cdot\mathrm{Vol}\left(B_p\left(x,\left(1-\frac{4}{(2t)^p}\right)^{\frac{1}{p}}\right)\right)\\ &= \left(1-\frac{3}{(2t)^p}\right)^{\frac{d}{p}}-2\cdot\left(1-\frac{4}{(2t)^p}\right)^{\frac{d}{p}}\\ &= \left(1-\frac{3}{(2t)^p}\right)^{\frac{d}{p}}\cdot\left(1-2\cdot\left(1-\frac{1}{(2t)^p-3}\right)^{\frac{d}{p}}\right)=\Omega(1)\cdot e^{-\frac{6}{2^p\cdot p}\cdot\frac{d}{t^p}}\quad ,\end{aligned}$$

where the last equality follows by the fact $1-x\ge e^{-2x}$, and as $t\ge 3$ and thus for large enough d, $2\cdot\left(1-\frac{1}{(2t)^p-3}\right)^{\frac{d}{p}}\le\frac{1}{2}$. As $\frac{\mathrm{Vol}(\mathrm{B_p}(\mathrm{x},1)\cup\mathrm{B_p}(\mathrm{x},1))}{\mathrm{V_p}}\le 2\cdot V_p$, the lemma follows. □

Observation 1. *For $p\in[1,2]$, and every pair of points $x,y\in\mathbb{R}^d$ at distance $\|x-y\|_p\le\frac{w_i}{4d}$, the probability that x and y belong to the same clusters in $\mathcal{C}_p^{w_i}$ is bounded by*

$$\frac{Vol_d\left(B_p(x,w_i)\cap B_p(y,w_i)\right)}{Vol_d\left(B_p(x,w_i)\cup B_p(y,w_i)\right)}\ge 1-4d\cdot\frac{\|x-y\|_p}{w_i}\quad .$$

Proof. By triangle inequality, for every $z\in\mathbb{R}^d$, $\|z-y\|_p\le\|z-x\|_p+\|x-y\|_p$, and hence $B_p\left(x,w_i-\|x-y\|_p\right)\subseteq B_p\left(x,w_i\right)\cap B_p\left(y,w_i\right)$. In particular,

$$\begin{aligned}&\mathrm{Vol}_d\left(B_p(x,w_i)\cap B_p(y,w_i)\right)\ge\mathrm{Vol}_d\left(B_p(x,w_i-\|x-y\|_p)\right)\\ &\qquad=\left(1-\frac{\|x-y\|_p}{w_i}\right)^d\cdot\mathrm{Vol}_d\left(B_p(x,w_i)\right)\ge\left(1-2d\cdot\frac{\|x-y\|_p}{w_i}\right)\cdot\mathrm{Vol}_d\left(B_p(x,w_i)\right)\quad ,\end{aligned}$$

where the last inequality holds as $1 - x \leq e^{-x} \leq 1 - \frac{x}{2}$. Denote $\mathrm{Vol}_d\left(B_p(x, w_i)\right) = Q$, and $\alpha \cdot Q = \mathrm{Vol}_d\left(B_p(x, w_i) \cap B_p(y, w_i)\right)$. Set $\beta = 1 - \alpha$, then $\beta \leq 2d \cdot \frac{\|x-y\|_p}{w_i}$. It follows that

$$\frac{\mathrm{Vol}_d\left(B_p(x, w_i) \cap B_p(y, w_i)\right)}{\mathrm{Vol}_d\left(B_p(x, w_i) \cup B_p(y, w_i)\right)} = \frac{\alpha \cdot Q}{2Q - \alpha \cdot Q} = \frac{1 - \beta}{1 + \beta} \geq 1 - 2\beta \geq 1 - 4d \cdot \frac{\|x - y\|_p}{w_i} \ .$$

□