Labeled Nearest Neighbor Search and Metric Spanners via Locality Sensitive Orderings

Arnold Filtser

Department of Computer Science, Bar-Ilan University, Ramat-Gan, Israel

Abstract

Chan, Har-Peled, and Jones [SICOMP 2020] developed locality-sensitive orderings (LSO) for Euclidean space. A \((\tau, \rho)-\text{LSO}\) is a collection \(\Sigma\) 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. \(\sigma\) are in the \(\rho\)-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, \(\ell_p\), 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.

2012 ACM Subject Classification Theory of computation \(\rightarrow\) Computational geometry; Theory of computation \(\rightarrow\) Sparsification and spanners

Keywords and phrases Locality sensitive ordering, nearest neighbor search, high dimensional Euclidean space, doubling dimension, planar and minor free graphs, path reporting low hop spanner, fault tolerant spanner, reliable spanner, light spanner

Digital Object Identifier 10.4230/LIPIcs.SoCG.2023.33


Funding This research was supported by the Israel Science Foundation (grant No. 1042/22).

1 Introduction

1.1 Locality Sensitive Ordering

Chan, Har-Peled, and Jones [30] 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 \(\{(\tau, \rho)\text{-LSO}\}\). Given a metric space \((X, d_X)\), we say that a collection \(\Sigma\) of orderings is a \((\tau, \rho)\)-LSO if \(|\Sigma| \leq \tau\), and for every \(x, y \in X\), there is a linear ordering \(\sigma \in \Sigma\) such that (w.l.o.g.) \(x \preceq_{\sigma} y\) and the points between \(x\) and \(y\) w.r.t. \(\sigma\) could be partitioned into two consecutive intervals \(I_x, I_y\) where \(I_x \subseteq B_X(x, \tau \cdot d_X(x, y))\) and \(I_y \subseteq B_X(y, \tau \cdot d_X(x, y))\). \(\rho\) is called the stretch parameter.

\(^1\) 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 \(\sigma\).
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. [30] constructed \((O_d(\epsilon^{-d}) \cdot \log \frac{1}{\epsilon}, \epsilon)\)-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 [27, 28] constructed reliable spanners using LSO, obtaining considerably superior results compared with previous techniques.

Filtser and Le [49] generalized Chan et al. [30] result to doubling spaces,\(^2\) showing that every metric space with doubling dimension \(d\) admits a \((\epsilon^{-O(d)}, \epsilon)\)-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 [49], we introduce the closely related notion of rooted-LSO, which has some additional structure. All the results and constructions for left-sided LSO in [49] hold for rooted LSO as well. We refer to [49] for a comparison between the different notions, and to Figure 1 for an illustration.

\[\begin{align*}
\text{Definition 2} & ((\tau, \rho)\text{-triangle-LSO}). \text{ Given a metric space } (X, d_X), \text{ we say that a collection } \\
\text{of orderings } \Sigma \text{ of orderings is a } (\tau, \rho)\text{-triangle-LSO if } |\Sigma| \leq \tau, \text{ and for every } x, y \in X, \text{ there is an ordering } \\
\sigma \in \Sigma \text{ such that } (\text{w.l.o.g.)} \ x \prec_\sigma y, \text{ and for every } a, b \in X \text{ such that } x \preceq_\sigma a \preceq_\sigma b \preceq_\sigma y \text{ it holds that } d_X(a, b) \leq \rho \cdot d_X(x, y).
\end{align*}\]

\[\begin{align*}
\text{Definition 3} & ((\tau, \rho)\text{-rooted-LSO}). \text{ Given a metric space } (X, d_X), \text{ we say that a collection } \\
\Sigma \text{ of orderings over subsets of } X \text{ is a } (\tau, \rho)\text{-rooted-LSO if the following hold:} \text{}} \\
\text{Each point } x \in X \text{ belongs to at most } \tau \text{ orderings in } \Sigma. \\
\text{Each ordering } \sigma \in \Sigma \text{ is associated with a point } x_\sigma \in X, \text{ which is the first in the order, and} \\
\text{such that the ordering is w.r.t. distances from } x_\sigma \text{ (i.e. } y \preceq_\sigma z \Rightarrow d_X(x_\sigma, y) \leq d_X(x_\sigma, z)). \\
\text{For every pair of points } u, v, \text{ there is some } \sigma \in \Sigma \text{ containing both } x, y, \text{ and such that } \\
d_G(u, x_\sigma) + d_G(x_\sigma, v) \leq \rho \cdot d_G(u, v).
\end{align*}\]

\(^2\) A metric \((X, d)\) has doubling dimension \(d\) if any ball of radius \(2r\) can be covered by \(2^d\) balls of radius \(r\).
Filtser and Le [49] 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.

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. ($^\ast$) $O_d$ hides an arbitrary function of $d$, the number of orderings in [30] LSO is $O_d(e^{-d}) \cdot \log \frac{1}{\epsilon} = 2^{O_d} \cdot d^{\frac{\epsilon^2}{2}} \cdot e^{-d} \cdot \log \frac{1}{\epsilon}$.

<table>
<thead>
<tr>
<th>LSO type</th>
<th>Metric Space</th>
<th># of orderings ($\tau$)</th>
<th>Stretch ($\rho$)</th>
<th>Ref</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Classic) LSO</td>
<td>Euclidean space $\mathbb{R}^d$</td>
<td>$O_d(e^{-d}) \cdot \log \frac{1}{\epsilon}$ ($^\ast$)</td>
<td>$\epsilon$</td>
<td>[30]</td>
</tr>
<tr>
<td></td>
<td>Doubling dimension $d$</td>
<td>$e^{-O_d} \cdot \log \frac{1}{\epsilon}$</td>
<td>$\epsilon$</td>
<td>[49]</td>
</tr>
<tr>
<td>Triangle-LSO</td>
<td>General metric</td>
<td>$O(n^{\frac{1}{2}} \cdot \log n \cdot \frac{\log k}{\epsilon})$</td>
<td>$2k + \epsilon$</td>
<td>[49]</td>
</tr>
<tr>
<td></td>
<td>Euclidean space $\mathbb{R}^d$</td>
<td>$e^{\frac{1}{2} \cdot \log \frac{1}{\epsilon}} \cdot O \left(d^{\frac{\epsilon^2}{2}} \cdot \log \frac{1}{\epsilon} \right)$</td>
<td>$(1 + \epsilon)t$ thm 4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Triangle LSO for $p \in [1, 2]$</td>
<td>$e^{O\left(\log n \right)} \cdot O(d)$</td>
<td>$t$ thm 5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Triangle LSO for $p \in [2, \infty]$</td>
<td>$O(d)$</td>
<td>$d^{1 - \frac{1}{p}}$ fulLIV[46]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Doubling dimension $d$</td>
<td>$2^{O(d)} \cdot d \cdot \log^2 t$</td>
<td>$t$ thm 6</td>
<td></td>
</tr>
<tr>
<td>Rooted LSO</td>
<td>Tree</td>
<td>$\log n$</td>
<td>1</td>
<td>[49]</td>
</tr>
<tr>
<td></td>
<td>Treewidth $k$</td>
<td>$k \cdot \log n$</td>
<td>1</td>
<td>[49]</td>
</tr>
<tr>
<td></td>
<td>Planar / fixed minor free</td>
<td>$O \left(\frac{1}{2} \cdot \log^2 n \right)$</td>
<td>$1 + \epsilon$</td>
<td>[49]</td>
</tr>
</tbody>
</table>

Previously constructed LSO for the Euclidean space [30], as well as for metric spaces with doubling dimension $d$ [49], 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 [30] one can only construct $(n, \epsilon)$-LSO (note that every metric admits $(1 + \delta)$-triangle LSO. In particular, for every set of $n$ points in $\mathbb{R}_d$, using the Johnson Lindenstrauss dimension reduction [61], for every fixed $t > 1$, we can construct $(n^{\frac{t}{2}} \cdot O \left(\log \frac{1}{\epsilon} \cdot n \right), O(t) \cdot (1 + \delta)t)$-triangle LSO, or $(O(\log n), O(\sqrt{\log n}))$-triangle LSO, a quadratic improvement compared with general $n$-point metric spaces!

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. Old and new LSO construction are summarized in Table 1.

Theorem 4. 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 $O \left(\frac{d^{\frac{1}{2}}}{\sqrt{t}} \cdot \log \left(\frac{2\sqrt{d}}{t} \right) \cdot \log \frac{d}{\delta} \cdot e^{\frac{1}{2} \cdot \log \frac{1}{\epsilon}} \cdot (1 + \delta)t \right)$-triangle LSO.

For $t = \frac{2}{3} \sqrt{d}$ and $\delta = \frac{1}{2}$, we obtain $O(d \log d, \sqrt{d})$-triangle LSO. In particular, for every set of $n$ points in $\mathbb{R}_d$, using the Johnson Lindenstrauss dimension reduction [61], for every fixed $t > 1$, we can construct $(n^{\frac{t}{2}} \cdot O \left(\log \frac{1}{\epsilon} \cdot n \right), O(t) \cdot (1 + \delta)t)$-triangle LSO, or $(O(\log n), O(\sqrt{\log n}))$-triangle LSO, a quadratic improvement compared with general $n$-point metric spaces!

This follows from a theorem by Walecki [7] who showed that the edges of the $K_n$ clique graph can be partitioned into $\left(\frac{1}{2} \right)$ Hamiltonian paths.
Interestingly, we show that the \( O(d \log d, \sqrt{d}) \)-triangle LSO \( \Sigma \) for \( \ell_2 \), is in the same time also a \( O(d \log d, d^{\epsilon}) \)-triangle LSO for \( \ell_p \) where \( p \in [1, 2] \), and \( O(d \log d, d^{-1+\epsilon}) \)-triangle LSO for \( \ell_p \) where \( p \in [2, \infty] \). For \( p \in [1, 2] \), we generalize Theorem 4 to \( \ell_p \) spaces to get the entire \#ordering-stretch trade-off. Finally, we generalize Theorem 4 to general metric spaces with doubling dimension \( d \).

\[ \text{Theorem 5.} \quad \text{For every } p \in [1, 2], t \in [5, d^{\epsilon}] \text{ and } d \geq 1, \text{ the } d \text{-dimensional } \ell_p \text{ space admits } \Theta^p(\epsilon) \cdot \tilde{O}(d, t) \text{-triangle LSO.} \]

\[ \text{Theorem 6.} \quad \text{Given a metric space } (X, d_X) \text{ with doubling dimension } d, \text{ and parameter } t \in [\Omega(1), d], X \text{ admits } 2^{O(q/t)} \cdot d \cdot \log^2 t, t \text{-triangle LSO.} \]

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 \( \ell_p \) spaces (see the survey [11]), and also in various norm spaces over \( \mathbb{R}^d \) (see e.g. [12, 13]). NNS data structures were also constructed beyond normed spaces. Some examples are Earth-Mover distance [60], Edit Distance [79, 11], and Fréchet distance [59, 41, 43, 47]. 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: \( d_{\mathbb{R}^d}((x_1, \ldots, x_d), (y_1, \ldots, y_d)) = \|(x_1, \ldots, x_d) - (y_1, \ldots, y_d)\|_2 \).

For general metric spaces, Krauthgamer and Lee [66] 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 below), 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.

\(^4\) An exact distance oracle \( D \) is a data structure that given two points \( x, y \), returns \( \text{est}(x, y) = d_X(x, y) \). A distance oracle of stretch \( t \) returns a value \( \text{est}(x, y) \) in \( [d_X(x, y), t \cdot d_X(x, y)] \).
An interesting case studied by Abraham, Chechik, Krauthgamer, and Wieder [3] 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 $\Delta$, Abraham et al. [3] 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 $O((\epsilon^{-1} \cdot \log \log n + t_{DO}) \cdot \log \log N + \log n \cdot \Delta \cdot t_{DO})$.

Linear dependence on the degree is a very limiting requirement, as planar graphs have a priori 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 [84, 64, 70]. Unfortunately, Abraham et al. [3] showed both assumptions to be necessary. Specifically, the dependence on the degree is necessary, as every NNS data structure with space at most $O(\frac{N}{\log n})$ must probe the distance oracle at least $\Omega(\Delta \log n)$ 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 $O(\frac{\log n}{\epsilon})$, $N \leq n^2$, 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 [66] and [3]) 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). On 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.

\textbf{Definition 7 (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) 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 $p \in P$: $d_X(p, q) \leq t \cdot \min_{x \in P} d_X(x, q)$. 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 try to imitate the natural hint provided by the name of the points themselves in metric spaces with implicit distance

\(^5\) After a long line of work, the state of the art (by Long and Pettie [73]) requires either super-linear space $N^{1+o(1)}$, or very large query time $N^{o(1)}$, both quite undesirable.

\(^6\) For example, consider 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 $P$. 

\textbf{A. Filtser 33:5}
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 $\ell_\infty$ [75, 1]. That is we assign each point $x$ a vector $v_x \in \mathbb{R}^d$ such that $\forall x, y \in X$, $d_X(x, y) \leq \|v_x - v_y\|_\infty \leq (2k-1) \cdot d_X(x, y)$, and use the vectors as labels. Given an $n$ point subset $P \subseteq X$ with its respective labels (vectors), use Indyk’s NNS [58] over $\{v_x\}_{x \in P}$ to construct a NNS data structure $D_{\text{Ind}}$ with approximation factor $O(\log_{1 + \frac{1}{k}} \log d) = O(t \cdot \log \log N)$ w.r.t. the $\ell_\infty$ vectors, space $\tilde{O}(d \cdot n^{1 + \frac{1}{k}}) = \tilde{O}(N^{\frac{1}{k}} \cdot n^{1 + \frac{1}{k}})$, and query time $\tilde{O}(n^{1 + \frac{1}{k}})$. Given a query $q$, we will simply query $D_{\text{Ind}}$ 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 showing that $(\tau, \rho)$-rooted LSO implies a labeled $\rho$-NNS with label size $O(\tau)$, space $O(n \cdot \tau)$, query time $\tilde{O}(\tau)$, and update time $\tilde{O}(\tau \cdot \log \log N)$. 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 [29]).

<table>
<thead>
<tr>
<th>Family</th>
<th>stretch</th>
<th>label</th>
<th>query time</th>
<th>update time</th>
<th>Ref</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minor free</td>
<td>$1 + \epsilon$</td>
<td>$O(\frac{1}{k} \log^2 N)$</td>
<td>$O(\frac{1}{k} \log^2 N)$</td>
<td>$\frac{1}{k} \cdot \tilde{O}(\log \log N)$</td>
<td>FullV[46]</td>
</tr>
<tr>
<td>Treewidth $k$</td>
<td>$1$</td>
<td>$O(k \log N)$</td>
<td>$O(k \log N)$</td>
<td>$k \cdot \tilde{O}(\log N)$</td>
<td>FullV[46]</td>
</tr>
<tr>
<td>Correlation $k$</td>
<td>$1 + \epsilon$</td>
<td>$O_{\xi}(\sqrt{N})$</td>
<td>$O_{\xi}(\sqrt{N})$</td>
<td>$O_{\xi}(\sqrt{N})$</td>
<td>FullV[46]</td>
</tr>
<tr>
<td>Ultrametric</td>
<td>$1$</td>
<td>$O(\log N)$</td>
<td>*</td>
<td>*</td>
<td>FullV[46]</td>
</tr>
<tr>
<td>General Metric</td>
<td>$8(1 + \epsilon)k$</td>
<td>$O(\log N \cdot \log N)$</td>
<td>$O(\frac{1}{k} \cdot *)$</td>
<td>$O(\frac{2}{k} N \cdot *)$</td>
<td>FullV[46]</td>
</tr>
<tr>
<td>$t &lt; 2k + 1$</td>
<td>$\Omega(N^{\frac{1}{k}})$</td>
<td>*</td>
<td>arbitrary</td>
<td>arbitrary</td>
<td>FullV[46]</td>
</tr>
<tr>
<td>Doubling $d$</td>
<td>$t$</td>
<td>$2^{O(\frac{\log N}{t})} \cdot \tilde{O}(d) \cdot \log N$</td>
<td>$2^{O(\frac{\log N}{t})} \cdot \tilde{O}(d) \cdot *$</td>
<td>$2^{O(\frac{\log N}{t})} \cdot \tilde{O}(d) \cdot *$</td>
<td>FullV[46]</td>
</tr>
</tbody>
</table>

Next, we prove a meta theorem, showing that $(\tau, \rho)$-triangle LSO implies a labeled $2\rho$-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 [11], and additional discussion in the full version [46]). 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 references in the full version [46]). 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 [75],[58] 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: $O(\epsilon^{-1} \cdot \log \log N)$. We show that the standard information theoretic bound applies for the labeled NNS as well, specifically, for...
stretch \( t < 2k + 1 \), the label size must be \( \tilde{O}(n^{\frac{2}{k}}) \) (regardless of query time). 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.

### 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 \( x, y \in X \). The classic parameter of study is the trade-off between stretch and sparsity (number of edges). Althöfer et al. [8] showed that every \( n \) point metric space admits a \( 2k - 1 \) spanner with \( O(n^{1 + \frac{k}{2}}) \) 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 [38, 52]. We refer to the book [77], and the survey [4] for an overview.

**Path Reporting Low Hop Spanners.** Recently, Kahalon, Le, Milenkovic, and Solomon [62] 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 [5, 23]. Low hop networks are also known to be more reliable [23, 87, 82], and used in electricity and telecommunications [23], and many other (practical) network design problems [71, 16, 55, 54, 81]. Hop-constrained network approximation is often used in parallel computing [36, 14], as the number of hops governs the number of required parallel rounds (e.g., in Dijkstra).

Kahalon et al. [62] constructed path reporting low-hop spanners for many spaces, such as path reporting \( 2 \)-hop \( O(k) \)-spanners with \( O(n^{1 + \frac{k}{2}}) \cdot k \cdot \log n \) edges, and \( O(1) \) query time for general metrics, and path reporting \( 2 \)-hop \( (1 + \epsilon) \)-spanners with \( O(\frac{2}{\epsilon^2} \cdot \log^2 n) \) 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, minimum weight trees (MST), and online MST verification.

**Our Contribution.** Kahalon et al. [62] 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

\[\text{\footnotesize{SoCG 2023}}\]
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, where there is an additional additive factor of \( O(\epsilon^{-2d}) \log \frac{1}{\epsilon} \).

<table>
<thead>
<tr>
<th>Metric family</th>
<th>stretch</th>
<th>sparsity</th>
<th>query time</th>
<th>Ref</th>
</tr>
</thead>
<tbody>
<tr>
<td>General Metric</td>
<td>( O(k) )</td>
<td>( O\left(n^{1+\frac{d}{2}} \cdot k \cdot \log n\right) )</td>
<td>( O(1) )</td>
<td>[62]</td>
</tr>
<tr>
<td>(1 + ( \epsilon ))(4k - 2)</td>
<td>( O(n^{1+\frac{d}{2}} \cdot k) )</td>
<td>( O(\epsilon^{-1} \cdot \log 2k) )</td>
<td>FullV[46], [85]</td>
<td></td>
</tr>
<tr>
<td>Doubling</td>
<td>( 1 + \epsilon )</td>
<td>( \epsilon^{-O(d)} \cdot n \cdot \log n )</td>
<td>( O(\epsilon^{-1} \cdot \log 2k) )</td>
<td>FullV[46]</td>
</tr>
<tr>
<td>Euclidean ( \mathbb{R}^d )</td>
<td>( 1 + \epsilon )</td>
<td>( O_d(\epsilon^{-d}) \cdot n \cdot \log n )</td>
<td>( O(d) )</td>
<td>FullV[46]</td>
</tr>
<tr>
<td>( \ell_p^d ), ( p \in [1,2] )</td>
<td>( t )</td>
<td>( O(d) \cdot \epsilon^{O(d)} \cdot \log n )</td>
<td>( O(d) )</td>
<td>FullV[46]</td>
</tr>
<tr>
<td>( \ell_p^d ), ( p \in [2, \infty] )</td>
<td>( 2 \cdot d^{1-\frac{d}{2}} )</td>
<td>( \hat{O}(d) \cdot n \cdot \log n )</td>
<td>( \hat{O}(d) )</td>
<td>FullV[46]</td>
</tr>
<tr>
<td>Tree</td>
<td>1</td>
<td>( O(n \cdot \log n) )</td>
<td>( O(1) )</td>
<td>[62]</td>
</tr>
<tr>
<td>Fixed</td>
<td>( 1 + \epsilon )</td>
<td>( O\left(n \cdot \epsilon^{-d} \cdot \log^2 n\right) )</td>
<td>( O(\epsilon^{-2} \cdot \log^2 n) )</td>
<td>[62]</td>
</tr>
<tr>
<td>Minor Free</td>
<td>( 1 + \epsilon )</td>
<td>( O\left(n \cdot \epsilon^{-d} \cdot \log^2 n\right) )</td>
<td>( O(\epsilon^{-1} \cdot \log n) )</td>
<td>FullV[46]</td>
</tr>
<tr>
<td>Planar</td>
<td>( 1 + \epsilon )</td>
<td>( O(n \cdot \epsilon^{-d} \cdot \log^2 n) )</td>
<td>( O(\epsilon^{-1}) )</td>
<td>FullV[46]</td>
</tr>
<tr>
<td>Treewidth ( k )</td>
<td>1</td>
<td>( O(n \cdot k \cdot \log n) )</td>
<td>( O(k) )</td>
<td>FullV[46]</td>
</tr>
</tbody>
</table>

Fault tolerant spanners. Levcopoulos, Narasimhan, and Smiid [72] 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 \( \forall u, v \in F, d_H(u, v) \leq t \cdot d_X(u, v) \). For general metrics, after a long line of work [34, 39, 20, 22, 40, 21, 80], it was shown that every \( n \)-vertex graph admits an efficiently constructible \( f \)-vertex-fault-tolerant \((2k - 1)\)-spanner with \( O(f^{1+k} \cdot n^{1+1/\ell}) \) edges, which is optimal assuming the Erdős–Girth Conjecture [44]. For \( n \)-points in \( d \) dimensional Euclidean space, or more generally in a space of doubling dimension \( d \), a fault tolerant spanner \((1 + \epsilon)\)-spanner were constructed with \( \epsilon^{-O(d)} \cdot f \cdot n \) edges [72, 74, 83].

Kahalon et al. [62] 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 [62] 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 \( \infty \)). It is natural to ask whether it is possible to construct such a spanner with only a linear dependence, and not quadratic as in [62].

Our Contribution. One can easily construct \( f \)-fault tolerant \( 1 \)-spanner for the path graph with \( O(n f \log n) \) edges. We observe that using \( O(n f \log n) \) 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 [6, 68]). 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 \(e^{-O(d)} \cdot f \cdot n \cdot \log n\) edges, getting the desired linear dependence on \(f\). See Table 4 for a summary of results.

### 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 the 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. [26] introduced the notion of a reliable spanner. A \(\nu\)-reliable spanner is a graph such that for every failure set \(B \subseteq X\), the residual spanner \(H \setminus B\) is a \(t\)-spanner for \(X \setminus B^+\), where \(B^+ \supseteq B\) is a superset of cardinality at most \((1 + \nu) \cdot |B|\). An oblivious \(\nu\)-reliable \(t\)-spanner is a distribution \(D\) over spanners, such that for every failure set \(B\), \(H \setminus B\) is a \(t\)-spanner for \(X \setminus B^+_H\), 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.

\(\nu\)-Reliable spanners were constructed for \(d\) dimensional Euclidean and doubling spaces with \(n \cdot e^{-O(d)} \cdot \tilde{O}(\log n)\) edges [27, 28, 49] by a reduction from (classical) LSO’s. Oblivious reliable spanners were constructed also for planar, minor free, treewidth graphs, and general metrics [49] by reductions from triangle, and rooted LSO’s (as well as from sparse covers [57]).

### Our Contribution

Our newly constructed triangle LSO’s for high dimensional Euclidean, \(\ell_p\) 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.
Labeled NNS and Metric Spanners via LSO’s

**Table 5** Summary of previous and new constructions of \( \nu \)-reliable spanners.

<table>
<thead>
<tr>
<th>Family</th>
<th>stretch</th>
<th>guarantee</th>
<th>size</th>
<th>ref</th>
</tr>
</thead>
<tbody>
<tr>
<td>Euclidean ( (\mathbb{R}^d, | \cdot |_2) )</td>
<td>( 1 + \epsilon )</td>
<td>Deterministic</td>
<td>( n \cdot O((e^{-\epsilon_d}) \nu^{-\epsilon} \cdot O(\log n)) )</td>
<td>[27]</td>
</tr>
<tr>
<td></td>
<td>( 1 + \epsilon )</td>
<td>Oblivious</td>
<td>( n \cdot O((e^{-\epsilon_d}) \cdot O(\nu^{-\epsilon} \cdot O(\log \log n)^2) )</td>
<td>[28]</td>
</tr>
<tr>
<td></td>
<td>( (1 + \epsilon)t )</td>
<td>Oblivious</td>
<td>( \nu^{-1} \cdot \epsilon e^{\nu \epsilon} \cdot (1 + \frac{\nu}{\epsilon}) \cdot O(n \cdot d^{\nu}) )</td>
<td>FullV[46]</td>
</tr>
<tr>
<td>( \ell_p^d ) for ( p \in [1, 2] )</td>
<td>( t )</td>
<td>Oblivious</td>
<td>( \nu^{-1} \cdot O(\log n \cdot d^2) )</td>
<td>FullV[46]</td>
</tr>
<tr>
<td>( \ell_p^d ) for ( p \in [2, \infty] )</td>
<td>( 2 \cdot d^{1 - \frac{1}{p}} )</td>
<td>Oblivious</td>
<td>( \nu^{-1} \cdot \nu^{-1} \cdot O(n \cdot d^2) )</td>
<td>FullV[46]</td>
</tr>
<tr>
<td>Doubling dimension ( d )</td>
<td>( 1 + \epsilon )</td>
<td>Deterministic</td>
<td>( n \cdot e^{-O(d)} \nu^{-6} \cdot O(\log n) )</td>
<td>[49]</td>
</tr>
<tr>
<td></td>
<td>( 1 + \epsilon )</td>
<td>Oblivious</td>
<td>( n \cdot e^{-O(d)} \nu^{-1} \log \nu^{-1} \cdot O(\log \log n)^2 )</td>
<td>[49]</td>
</tr>
<tr>
<td></td>
<td>( t )</td>
<td>Oblivious</td>
<td>( O(n \cdot \nu^{-1}) \cdot 2^{O(d \epsilon / \nu)} )</td>
<td>FullV[46]</td>
</tr>
<tr>
<td>General metric</td>
<td>( 8t + \epsilon )</td>
<td>Oblivious</td>
<td>( O(n^{1 + \frac{1}{2p}} \cdot \epsilon^{- \nu}) \cdot \nu^{-1} )</td>
<td>[49]</td>
</tr>
<tr>
<td>Tree</td>
<td>( 2 )</td>
<td>Oblivious</td>
<td>( n \cdot O(\nu^{-1} \log^2 n) )</td>
<td>[49]</td>
</tr>
<tr>
<td>Planar/Minor-free</td>
<td>( 2 + \epsilon )</td>
<td>Oblivious</td>
<td>( n \cdot O(\nu^{-1} \epsilon^{-2} \log^2 n) )</td>
<td>[49]</td>
</tr>
</tbody>
</table>

**Table 6** Summary of previous and new results of light spanners for high dimensional metric spaces. Interestingly, for \( p \in [1, 2] \) [49] obtain lightness \( O\left(\frac{1 + \epsilon_p}{\log n} \cdot n \cdot \log^2 \cdot \log^2 \right) \) regardless of dimension, which is superior to ours for \( d \gg \log n \).

<table>
<thead>
<tr>
<th>Metric space</th>
<th>Stretch</th>
<th>Lightness</th>
<th>Ref</th>
</tr>
</thead>
<tbody>
<tr>
<td>Euclidean space</td>
<td>( O(t) )</td>
<td>( O(n^{3/4} \cdot \log n \cdot t) )</td>
<td>[69]</td>
</tr>
<tr>
<td></td>
<td>( O(t) )</td>
<td>( O(e^{\nu \epsilon} \cdot \log^2 n \cdot t) )</td>
<td>[50]</td>
</tr>
<tr>
<td></td>
<td>( (1 + \epsilon)2t )</td>
<td>( e^{\nu \epsilon} \cdot (1 + \frac{\nu}{\epsilon}) \cdot O\left(\frac{d^{\nu}}{\epsilon} \cdot \log n\right) )</td>
<td>FullV[46]</td>
</tr>
<tr>
<td></td>
<td>( (1 + \epsilon)4t )</td>
<td>( e^{\nu \epsilon} \cdot (1 + \frac{\nu}{\epsilon}) \cdot O\left(\frac{d^{\nu}}{\epsilon} \cdot \log^2 n\right) )</td>
<td>FullV[46]</td>
</tr>
<tr>
<td>Doubling dimension</td>
<td>( O(t) )</td>
<td>( O(2^{\nu} \cdot t \cdot \log^2 n) )</td>
<td>[50]</td>
</tr>
<tr>
<td></td>
<td>( O(t) )</td>
<td>( 2^{O(\epsilon / \nu)} \cdot d \cdot \log^2 t \cdot \log^2 n )</td>
<td>FullV[46]</td>
</tr>
<tr>
<td></td>
<td>( d )</td>
<td>( O(d \cdot \log^2 n) )</td>
<td>[50]</td>
</tr>
<tr>
<td></td>
<td>( d )</td>
<td>( O(d \cdot \log^2 d \cdot \log^2 n) )</td>
<td>FullV[46]</td>
</tr>
<tr>
<td>( \ell_p^d ) for ( p \in [1, 2] )</td>
<td>( t )</td>
<td>( O\left(\frac{1 + \epsilon_p}{\log^{d+1} n} \cdot \log n \right) )</td>
<td>[50]</td>
</tr>
<tr>
<td>( \ell_p^d ) for ( p \in [2, \infty] )</td>
<td>( 2 \cdot d^{1 - \frac{1}{p}} )</td>
<td>( O\left(\log \cdot \log^2 \cdot \log^2 \right) )</td>
<td>FullV[46]</td>
</tr>
</tbody>
</table>

Obtaining light spanners for general graphs has been the subject of an active line of work [8, 31, 42, 18, 35, 51], where the state of the are by Le and Solomon [69] who obtained \( (1 + \epsilon)(2k - 1) \) spanner with lightness \( O(\epsilon^{-1} \cdot n^{2}) \). Light spanners were also studied extensively in Euclidean spaces (see the book [77]), doubling spaces [53, 51, 25], planar and minor free graphs [63, 65, 24, 67, 37], and high dimensional Euclidean and doubling spaces [56, 50, 69].

**Our Contribution.** Recently Le and Solomon [69] 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 $\xi_i$ (for some large $\xi$) into clusters of diameter $\xi$, such that close-by points are likely to be clustered together. In the created ordering $\sigma$, points in each cluster will be ordered consecutively and recursively. In particular, the ordering $\sigma$ 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 $\sigma$, they have to be clustered together in all the distance scales $\xi_i \geq t \cdot \|x - y\|_2$.

Our space partition in each scale is done using ball carving (ala [10]): pick a uniformly random series of centers $z_1, z_2, \ldots$. Each points is assigned to the cluster of the first center at distance at most $R = \frac{1}{2} \cdot \xi$. 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 then equal to the ratio between the volumes of intersection and union of balls: $\Pr[x, y \text{ clustered together}] = \frac{\text{Vol}_d(B(x, R) \cap B(y, R))}{\text{Vol}_d(B(x, R) \cup B(y, R))} \geq \Omega\left(\frac{1}{R}\right) \cdot \left(1 - \left(\frac{\|x - y\|_2}{R}\right)^2\right)^{d/2}$. We bound this ratio for the case $\|x - y\|_2 \leq \frac{R}{\sqrt{d}}$ using a lemma from [33]. For the general case, we prove that the ratio between these volumes is at least $\Omega\left(\frac{R}{\sqrt{d} |\|p - q\||_2}\right) \cdot \left(1 - \left(\frac{|\|p - q\||_2}{R}\right)^2\right)^{d/2}$, slightly improving a similar fact from [9], 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 $2\ell$). The improved analysis of the volumes ratio is significant for the $O(\sqrt{d})$-stretch regime, improving the number of orderings to $O(d)$ (compered with $O(d^{1.5})$ orderings if we were using [9]).

To generalize this construction to $\ell_p$ spaces, we use the exact same construction, replacing $\ell_2$ balls with $\ell_p$ balls. The volume ratio lemma from [32] for close-by points is replaced by a crude observation without any significant consequences to the resulting number of orderings. For the general case, we directly analyze the ratio of volumes for $\ell_p$-balls (our computation is similar to [78]). The rest of the analysis is the same.

Triangle LSO for doubling spaces. Ultrametrics are trees with additional structure, where each ultrametric admits a $(1,1)$-triangle LSO. $(\tau, \rho)$-ultrametric cover of a metric space $(X, d_X)$ is a collection $\mathcal{U}$ of $\tau$ ultrametrics such that every pair $x, y \in X$ is well approximated by the ultrametrics: $d_X(x, y) \leq \min_{\rho \in \mathcal{U}} d_{\mathcal{U}}(x, y) \leq \rho \cdot d_X(x, y)$. Filtser and Le [49] showed that $(\tau, \rho)$-ultrametric cover implies $(\tau, \rho)$-triangle LSO. We construct $(2^{O(\sqrt{d})} \cdot d \cdot \log^2 t, t)$-ultrametric cover for spaces with doubling dimension $d$, implying Theorem 6.

Our starting point for constructing the ultrametric cover is Filtser’s [45] padded partition cover, which is a collection of $\approx 2^{O(\sqrt{d})}$ space partitions where all clusters are of diameter at most $\Delta$, and every ball of radius $\frac{\Delta}{\tau}$ is fully contained in a single cluster in one of the partitions. We take a single partition from each distance scale, where the gap between the distance scales is somewhat large: $\Omega(\frac{1}{\tau})$. Initially these partitions are unrelated, and we “force” them to be laminar, 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 $(\tau, \rho)$ 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 [86], it only remains to choose one of the $2\tau$ 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. 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 $\sigma$ it belongs to, and the distance to the first point $x_\sigma$ (w.r.t. $d_X$). Given a query $q$, for each ordering $\sigma$ 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 $y_\sigma$ 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, $N^k$, which results in similar NNS label size, and query time (following the approach above). While the NNS label essentially cannot be improved, we can significantly reduce the query time. Our solution is to use Ramsey trees [19, 76, 17, 2], which are a collection of embeddings into ultrametrics $U$ such that each point $x$ has a single home ultrametric $U_x \in U$ 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 [48], 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 casing assumption (either polynomial aspect ratio or small failure probability) to even $O(1)$ label size.

**Path reporting low hop spanners.** A $(\tau, \rho)$-tree cover is similar to ultrametric cover discussed above, where the ultrametrics are replaced by trees. Kahalon et al. [62] 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 [62], 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 [85] 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 [15] 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. [62] 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.

References

Labeled NNS and Metric Spanners via LSO's


33:16 Labeled NNS and Metric Spanners via LSO’s


62 Onur Kahalon, Hung Le, Lazar Milenkovic, and Shay Solomon. Can’t see the forest for the trees: Navigating metric spaces by bounded hop-diameter spanners. In Alessia Milani and
Labeled NNS and Metric Spanners via LSO’s


