# Tree-Like Shortcuttings of Trees

Hung Le[*] Lazar Milenković[†] Shay Solomon[‡] Cuong Than[§]

**Abstract**

Sparse *shortcuttings* of trees—equivalently, sparse 1-spanners for tree metrics with bounded *hop-diameter*—have been studied extensively (under different names and settings), since the pioneering works of [Yao82, Cha87, AS87, BTS94], initially motivated by applications to range queries, online tree product, and MST verification, to name a few. These constructions were also lifted from trees to other graph families using known low-distortion embedding results. The works of [Yao82, Cha87, AS87, BTS94] establish a tight tradeoff between *hop-diameter* and sparsity (or average degree) for tree shortcuttings and imply constant-hop shortcuttings for $n$-node trees with sparsity $O(\log^* n)$. Despite their small sparsity, **all known constant-hop shortcuttings contain dense subgraphs** (of sparsity $\Omega(\log n)$), which is a significant drawback for many applications.

We initiate a systematic study of constant-hop tree shortcuttings that are "tree-like". We focus on two well-studied graph parameters that measure how far a graph is from a tree: *arboricity* and *treewidth*. Our contribution is twofold.

- **New upper and lower bounds for tree-like shortcuttings of trees**, including an *optimal* tradeoff between hop-diameter and treewidth for all hop-diameter up to $O(\log \log n)$. We also provide a lower bound for larger values of $k$, which together yield hop-diameter $\times$ treewidth $= \Omega((\log \log n)^2)$ *for all values of hop-diameter*, resolving an open question of [FL22b, Le23].
- **Applications of these bounds**, focusing on low-dimensional Euclidean and doubling metrics. A seminal work of Arya et al. [ADM+95] presented a $(1+\epsilon)$-spanner with constant hop-diameter and sparsity $O(\log^* n)$, but with large arboricity. We show that constant hop-diameter is sufficient to achieve arboricity $O(\log^* n)$. Furthermore, we present a $(1+\epsilon)$-stretch routing scheme in the *fixed-port* model with 3 hops and a local memory of $O(\log^2 n/ \log \log n)$ bits, resolving an open question of [KLMS22].

## 1 Introduction

Given a tree $T = (V, E)$ and an integer $k \geq 1$, a *tree shortcutting* of $T$ with hop-diameter $k$ is a graph $G = (V, E')$ such that for every two vertices $u, v \in V$, there is a path in $G$ consisting of at most $k$ edges such that $d_G(u, v) = d_T(u, v)$, where $d_G(u, v)$ (resp., $d_T(u, v)$) represents the distance between $u$ and $v$ in $G$ (resp., $T$). Besides achieving small (ideally constant) hop-diameter, the most basic parameter of a tree shortcutting is its number of edges, $|E'|$. The problem of constructing sparse tree shortcuttings with small hop-diameter has been

[*]University of Massachusetts Amherst. Email: hungle@cs.umass.edu.
[†]Tel Aviv University. Email: milenkovic.lazar@gmail.com.
[‡]Tel Aviv University. Email: solo.shay@gmail.com.
[§]University of Massachusetts Amherst. Email: cthan@umass.edu.

studied extensively (under different names and settings), since the pioneering works of [Yao82, Cha87, AS87, BTS94]. The first tradeoff was given by Yao [Yao82], who studied the problem of computing range queries on paths. Later, Chazelle [Cha87] showed how to extend these tradeoffs from paths to arbitrary trees. Few years after, Alon and Schieber [AS87] studied the problem of computing semigroup product along paths and trees. Given a tree $T$ whose vertices are elements of a semigroup, the goal is to preprocess it into a succinct data structure so that every subsequent semigroup product query can be answered using as few as possible operations. In the terminology of tree shortcutting, they showed that for any $n$-point path, one can get a shortcutting with hop-diameter $k$ and $O(n\alpha_k(n))$ edges. Here, $\alpha_k(n)$ is a very slowly-growing inverse Ackermann function defined in Section 2. In particular, their tradeoff implies that one can get hop-diameters 2, 3, and 4, with $O(\log n)$, $O(\log\log n)$, and $O(n\log^* n)$ edges, respectively. For trees, their hop-diameter grows by a factor of 2: their tree shortcutting achieves hop-diameter $2k$ using $O(n\alpha_k(n))$ edges. They also showed that for a line metric the tradeoff is tight. Some of the applications of shortcuttings mentioned in [AS87] include finding the max-flow in a multiterminal network, verifying MST, and updating an MST after increasing the cost of one of its edges. The problem of problem of achieving an optimal tradeoff between hop-diameter and sparsity was settled by Bodlaender et al. [BTS94], who gave an optimal tradeoff for trees matching the lower bound of [AS87] for paths.

These constructions were successfully used in various applications and were also lifted to other graph families. Most notably perhaps, a seminal work of Arya et al. [ADM+95] shows that every $n$-point Euclidean metric with a constant dimension admits a $(1+\epsilon)$-spanner with hop-diameter $k$ and $O_\epsilon(n\alpha_k(n))$ edges. Spanners are formally defined in Section 2. Roughly speaking, spanners generalize the notion of tree shortcutting: Given a tree $T$, a tree shortcutting of $T$ is a stretch-1 spanner of the tree metric induced by $T$. Filtser and Le [FL22b] used tree shortcutting to achieve low-treewidth embedding of planar graphs. (See [CFKL20, CLPP23, CCC+25] for additional results on low-treewidth embedding of planar and minor-free graphs and metrics.) In particular, the central part of the embedding result of [FL22b] is a low-treewidth shortcutting for trees. Another example is the compact routing scheme by Kahalon et al. [KLMS22]. (See Section 2 for a definition of compact routing.) Their routing scheme operates on top a network which is a low-treewidth tree shortcutting with hop-diameter 2.

Once the shortcutting has been computed, it serves as a "proxy" overlay network, on which the computation can proceed, which gives rise to huge savings in a number of quality measures, including global and local space usage, as well as in various notions of running time, which change from one setting to another. In some applications where limiting the hop-distances of paths is crucial, such as in some routing schemes, road and railway networks, and telecommunication, we might need to minimize the hop-distances; for example, imagine a railway network, where each hop in the route amounts to switching a train – how many of us would be willing to use more than, say, 4 hops? Likewise, what if each hop amounts to traversing a traffic light, wouldn't we prefer routes that minimize the number of traffic lights? In such cases, the designer of the system, or its users, might not be content with super-constant hop-distances, or even with a large constant, and it might be of significant value to achieve as small as possible hop-distances. Motivated by such practical considerations, we are primarily interested in values of hop-diameter $k$ that "approach" 1, mainly $k = 2, 3, 4, ...$ as there is no practical need in considering larger values of $k$.

The fundamental drawback of sparse tree shortcuttings is that they have dense subgraphs. In particular, all the aforementioned tree shortcutting constructions suffer from not being uniformly-sparse in the regime of constant $k$ — they have subgraphs with average degree of $\Omega(\log n)$. This lower bound on average degree is true for hop-diameter $k = 2$ and does not reduce with hop-diameter. Motivated by this, we initiate a systematic study of constant-hop tree shortcuttings that are "tree-like". In particular we are interested in two notions that capture uniform sparsity: *treewidth* and *arboricity*. (Formal definitions are given in Section 2.) These two graph families are prime examples of families admitting more efficient algorithms, in contrast to just sparse graphs. The key question underlying this work is the following.

**Question 1.** Given an $n$-vertex tree, is there a shortcutting with constant hop diameter $k$ with arboricity/treewidth of $o(\log n)$, and ideally close to constant? In particular, is it possible to achieve this using $k = 3, 4, ...$?

Getting below the $\log n$ bound is impossible for $k = 2$ due to the sparsity lower bound for shortcuttings with hop-diameter 2. A well-known construction achieves a bound of $O(\log n)$ on sparsity, treewidth, and arboricity in this regime. Also, with $\Theta(\log n)$ hops, one can get a constant bound on maximum degree and thus also on sparsity, treewidth, and arboricity [SE14]. We note, however, that the key focus of this paper is in the regime where the hop-diameter is constant. This is arguably the most important regime for all applications. A key insight of this work is that one can break the logarithmic bound, even for treewidth, by using as few as 3 hops. Building on this insight, we derive general nontrivial tradeoffs between the hop-diameter and the treewidth/arboricity, and demonstrate their applicability.

The aforementioned low-treewidth embedding result of Filtser and Le [FL22b] gives a tree shortcutting with hop-diameter $k = O(\log\log n)$ and treewidth $t = O(\log\log n)$. Using this result, they give a low-treewidth embedding of planar graphs into low-treewidth graphs. In particular, they show that if one can construct a tree shortcutting with hop-diameter $k$ and treewidth $t$, then one can embed a planar graph with diameter $D$ into a graph with treewidth $O(k \cdot t/\epsilon)$ with an additive distortion of $O(\epsilon D)$. Improving the product of $k \cdot t$ would immediately improve the bound on the treewidth of the embedding. The following question is left open in their work.

**Question 2** ([Le23, FL22b])**.** Is there a tree shortcutting with treewidth $t$ and hop-diameter $k$ such that $k \cdot t = o((\log\log n)^2)$? Furthermore, is there such a tree shortcutting with a constant hop-diameter?

The compact routing scheme of Kahalon et al. [KLMS22] achieves memory bound of $O(\log^2 n)$ bits for routing on tree shortcuttings with hop-diameter 2. The reason why they achieve a bound of $O(\log^2 n)$ bits is essentially due to the fact that the shortcutting has arboricity/treewidth of $\Theta(\log n)$. The main obstacle in improving their memory bound lies in understanding tree shortcuttings with small treewidth/arboricity. Breaking the bound of $\Theta(\log^2 n)$ bits is a main open question left in their work. Quoting [KLMS22]: "Whether or not one can use a spanner[1] of larger (sublogarithmic and preferably constant) hop-diameter for designing compact routing schemes with $o(\log^2 n)$ bits is left here as an intriguing open question."

**Question 3** ([KLMS22])**.** Given an $n$-vertex tree $T$, is there a compact routing scheme (operating on a shortcutting of $T$) with stretch 1 which uses $o(\log^2 n)$ bits of space for every node and achieves an $o(\log n)$, and ideally constant hop-diameter?

### 1.1 Our contribution

Perhaps the main contribution of this work is a conceptual one—identifying the importance of Question 1. In Section 1.1.1 we present new upper and lower bounds for tree shortcuttings, which answer Question 1 as well as Question 2 in the regime of hop-diameter $O(\log\log n)$. In Section 1.1.2 we present some extensions and applications of these bounds, which in particular settle Question 3.

[1] tree shortcutting in our terminology

### 1.1.1 Bounds for tree shortcuttings

We provide a thorough investigation of Question 1. First, we show that one can break the $\log n$ barrier on treewidth already for hop-diameter 3. Theorem 1.1 provides a general upper bound tradeoff between the hop-diameter and the treewidth, for all values of hop-diameter $k = O(\log\log n)$.

**Theorem 1.1.** *For every $n \geq 1$ and every $k = O(\log\log n)$, every $n$-vertex tree admits a shortcutting with hop-diameter $k$ and treewidth $O(k \log^{2/k} n)$ for even $k$ and $O(k(\frac{\log n}{\log\log n})^{2/(k-1)})$ for odd $k \geq 3$.*

**Remark.** It is impossible to extend the result of Theorem 1.1 to basic graph families, such as planar graphs or Euclidean metrics. See Section 1.2 for more details.

We also prove a lower bound tradeoff between the hop-diameter and the treewidth that matches the upper bound of Theorem 1.1 for all values of $k = O(\log\log n)$. Furthermore, we provide a lower bound for larger values of $k$, which together settle negatively Question 2, and in particular imply that the construction of [FL22b] with hop-diameter and treewidth both bounded by $O(\log\log n)$ is asymptotically optimal.

**Theorem 1.2.** *For every $n \geq 1$, every shortcutting with hop-diameter $k$ for an $n$-point path must have treewidth:*

- *$\Omega(k \log^{2/k} n)$ for even $k$ and $\Omega(k(\frac{\log n}{\log\log n})^{2/(k-1)})$ for odd $k \geq 3$, whenever $k \leq \frac{2}{\ln(2e)} \ln\log n$;*
- *$\Omega((\log\log n)^2/k)$ whenever $k > \frac{2}{\ln(2e)} \ln\log n$.*

The construction of Theorem 1.1 for $k = 3$ has treewidth, and thus also arboricity, bounded by $O(\log n/\log\log n)$. We next show that the bound on the arboricity can further be improved. In particular one can get rid of the factor $k$ in the tradeoff from Theorem 1.1 by introducing some slack to the exponent of $\log n$. In particular, we prove the following theorem.

**Theorem 1.3.** *For every two integers $n \geq 1$ and $k \geq 1$ and every $n$-vertex tree $T$, there is a shortcutting with hop-diameter $k$ and arboricity $O(\log^{12/(k+4)} n)$. Moreover, when the height of the tree is $h$, then the arboricity is $O(h^{6/(k+4)})$.*

Finally, we show an even better tradeoff between the hop-diameter and arboricity on paths.

**Theorem 1.4.** *For every $n \geq 1$ and every even $k \geq 2$, every $n$-point path admits a shortcutting with hop-diameter $k$ and arboricity $O(\alpha_{k/2+1}(n))$.*

In particular, Theorem 1.4 shows that one can get arboricity $O(\log\log n)$ with $k = 4$ and $O(\log^* n)$ with $k = 6$. The tradeoff is asymptotically tight, due to the sparsity lower bound of [AS87], as the arboricity of any graph is at most its average degree (up to a factor of 2).

We note that all our shortcutting constructions can be implemented in time linear in their size. We will skip the implementation details as this is not the main focus of this work.

### 1.1.2 Extensions and applications

We next extend Theorem 1.4, which provides a 1-spanner for line metrics with bounded hop-diameter and arboricity, to arbitrary doubling metrics, by increasing the stretch to $1 + \epsilon$.

**Theorem 1.5.** *Let $k$ be an even integer and let $\epsilon \in (0, 1/6)$ be an arbitrary parameter. Then, for every positive integer $n$, every $n$-point metric with doubling dimension $d$ admits a $(1+\epsilon)$-spanner with hop-diameter $k$ and arboricity $\epsilon^{-O(d)}\alpha_{k/2+1}(n)$.*

This significantly strengthens the construction of Arya et al. [ADM+95], by providing a uniformly sparse (rather than just sparse) construction with constant hop-diameter. Specifically, for hop-diameter $k$, we transition from sparsity $\epsilon^{-O(d)}\alpha_k(n)$ to arboricity $\epsilon^{-O(d)}\alpha_{k/2+1}(n)$; in particular, we get arboricity $O(\log^* n)$ with an hop-diameter of 6.

Using the results in tree covers [CCL+23, CCL+24], we can lift this tradeoff for planar and minor-free metrics.

**Theorem 1.6.** *Let $k$ be an even integer and let $\epsilon \in (0, 1/6)$ be an arbitrary parameter. Then, for every positive integer $n$, every $n$-point $K_h$-minor-free metric admits a $(1+\epsilon)$-spanner with hop-diameter $k$ and arboricity $O(\epsilon^{-3} \cdot 2^{h^{O(h)}/\epsilon} \cdot \alpha_{k/2+1}(n))$. Furthermore, if the metric is planar, one can construct such a spanner with arboricity $O(\epsilon^{-3} \cdot \alpha_{k/2+1}(n))$.*

Using the results in [BFN19, MN07], we obtain the following theorem for spanners of general metrics.

**Theorem 1.7.** *Let $k$ be an even integer, $r \geq 1$ be an arbitrary parameter. Then, for every positive integer $n$, every $n$-point metric $M_T$ admits an $O(n^{1/r} \log^{1-1/r} n)$-spanner with hop-diameter $k$ and arboricity $O(r \cdot \alpha_{k/2+1}(n))$. Alternatively, $M_T$ admits an $O(r)$-spanner with hop-diameter $k$ and arboricity $O(r \cdot n^{1/r} \cdot \alpha_{k/2+1}(n))$.*

In Section 5, we use the construction from Theorem 1.1 to devise a constant-hop routing scheme for tree metrics with stretch 1 and $O(\log^2 n/\log\log n)$ bits per vertex. We note that our routing algorithm operates on top of a network with hop-diameter 3 and treewidth $O(\log n/\log\log n)$.

**Theorem 1.8.** *For every $n$ and every $n$-vertex tree $T$, there is a 3-hop routing scheme in the fixed-port model for the metric $M_T$ induced by $T$ with stretch 1 that uses $O(\log^2 n/\log\log n)$ bits per vertex.*

This answers in the affirmative Question 3, which is the main open question from [KLMS22]. We show that the bound from Theorem 1.8 on memory per vertex is the best one could get in tree metrics, regardless of the number of hops! In particular, we prove the following theorem.

**Theorem 1.9.** *There is an infinite family of trees $T_n$, $n > 0$, such that any labeled fixed-port routing scheme with stretch 1 on a metric induced by $T_n$ has at least one vertex with total memory of $\Omega(\log^2 n/\log\log n)$ bits.*

Using known tree cover constructions, we obtain a 3-hop $(1+\epsilon)$-stretch routing in doubling metrics. Given a tree cover, we construct a routing scheme on top of each of the trees in the cover. The key challenge is in obtaining a mechanism for efficiently (without increasing the memory usage) determining which tree to use for routing between any two given source and target points. We focus here on doubling metrics, where such a mechanism is already given [CCL+25], and we use it as a black-box to derive our result. However, this is a general framework that can be applied to any other metric family that admits a tree cover construction, provided that one has the mechanism to efficiently determine the right tree on top of which the routing proceeds.

**Theorem 1.10.** *For every $n$ and every $n$-point metric with doubling dimension $d$, there is a 3-hop routing scheme with stretch $(1+\epsilon)$ that uses $O_{\epsilon,d}(\log^2 n/\log\log n)$ bits per vertex.*

This result provides the first routing scheme in Euclidean and doubling metrics, where the number of hops is $o(\log n)$, let alone as small as 3, and the labels consist of $o(\log^2 n)$ bits.

### 1.2 A natural limitation of low-treewidth shortcuttings and spanners

Theorem 1.1 provides an upper bound tradeoff between the hop-diameter and treewidth of shortcuttings for *trees*. Low-treewidth shortcuttings do not exist in most other basic graph families, such as planar graphs and Euclidean and doubling metrics, and this limitation holds even regardless of any hop-diameter bound. Indeed, the $\sqrt{n}$ by $\sqrt{n}$ unweighted grid is a planar graph of treewidth $\sqrt{n}$, and any shortcutting of the grid must include the entire grid. Similarly, for the doubling metric induced by the grid, as well as for the Euclidean space induced by a 2-dimensional grid point set, any shortcutting must contain the entire grid. In fact, a similar limitation applies even when considering low-treewidth spanners of stretch larger than 1 (again even regardless of their hop-diameter). In particular, it was observed in [DFG08] that for any planar graph of treewidth $k$, any $t$-spanner must have treewidth $\Omega(k/t)$, and a similar lower bound on the spanner treewidth was extended to the families of bounded genus and apex-minor-free graphs. Building on these lower bounds, fixed parameter tractability results for these graph families were given in [DFG08] (refer also to [FGvL11] for results for bounded degree graphs). Low-treewidth Euclidean spanners were studied by Buchin et al. [BRS25], who showed that any set of $n$ points in the Euclidean plane admits a $t$-spanner of treewidth $O(n/t)$, and this tradeoff between stretch and treewidth is asymptotically optimal. Corneil et al. [CDKX25] showed that for any constants $w \geq 2$ and $c \geq 1$ there exist graphs of treewidth $w$, such that no spanning subgraph of treewidth $w-1$ can be an *additive* $c$-spanner of such a graph.

## 2 Preliminaries

**Treewidth.** Given a graph $G$, let $V(G)$ and $E(G)$ be the vertex and edge set of $G$. A tree decomposition of $G$ is a pair $(X, T)$ where $T$ is a tree and $X = \{X_1, X_2, \ldots, X_l\}$ is a set of subsets of $V(G)$, called bags, associated with nodes in $T$ such that:

- Every vertex in $V(G)$ is in at least one bag in $X$.
- For every edge $e$ in $E(G)$, there is at least one bag containing both endpoints of $e$.
- For every vertex $u$ in $V(G)$, the bags containing $u$ in $X$ inducing a subtree of $T$.

The width of $(X, T)$ is $\max_{X_i \in X} |X_i| - 1$. The treewidth of $G$ is the minimum width over all the tree decomposition of $G$.

The *treewidth* of a graph is closely related to the size of the complete graph minors it contains. Recall that a *minor* of a graph $G$ is any graph obtained from $G$ by a sequence of vertex deletions, edge deletions, and edge contractions. Let $K_h$ denote the complete graph on $h$ vertices. It is widely known that treewidth is monotone under taking minors. Then, we obtain the following fundamental fact:

**Fact 2.1.** *If a graph $G$ contains $K_h$ as a minor, then the treewidth of $G$ is at least $h-1$.*

**Arboricity.** The *arboricity* of a graph $G$ measures how sparse its subgraphs can be. Formally, the arboricity of $G$, denoted by $\mathrm{arb}(G)$, is defined as

$$\mathrm{arb}(G) \;=\; \max_{H \subseteq G,\, |V(H)| \geq 2} \left\lceil \frac{|E(H)|}{|V(H)| - 1} \right\rceil,$$

where the maximum is taken over all subgraphs $H$ of $G$ with at least two vertices. Nash-Williams proved that the arboricity equal to the minimum number of forests that $G$ can decomposed into [NW64].

**Spanner.** A *t-spanner* of a graph $G$ is a spanning subgraph $H$ that approximately preserves distances between all pairs of vertices. Formally, for every $u, v \in V(G)$, $d_H(u,v) \le t \cdot d_G(u,v)$. A $t$-spanner of a metric space $(X, \delta)$ is a $t$ spanner of $\left(X, \binom{X}{2}, \delta\right)$. A spanner of a metric space is often known as geometric spanner.

**Tree cover.** A *tree cover* of a graph $G$ with stretch $t$ is a collection of spanning trees $\mathcal{T}$ of $G$ such that, for every $u, v \in V(G)$:

- The distance between $u$ and $v$ in any tree $T \in \mathcal{T}$ is at least the distance in $G$, i.e., $d_T(u,v) \ge d_G(u,v)$.
- There exists a tree $T \in \mathcal{T}$ such that $d_T(u,v) \le t \cdot d_G(u,v)$.

Similar to geometric spanner, a tree cover of a metric space $(X, \delta)$ is a tree cover of the complete graph $\left(X, \binom{X}{2}, \delta\right)$.

**Compact routing scheme.** A *routing scheme* is a distributed algorithm that, given a packet with a designated source and destination, specifies how the packet is forwarded along the edges of the network. Each node in the network has its own *routing table*, which stores local routing information, as well as a unique *label*. During a preprocessing phase, the network is initialized so that each node is assigned a routing table and a label. In the *labeled model*, labels are chosen by the designer (typically of size $\text{poly}(\log n)$), while in the *name-independent model*, labels are chosen adversarially.

Each packet carries a *header* containing the label of the destination and possibly additional auxiliary information. When forwarding a packet destined for vertex $v$, a node $u$ consults its routing table together with the label of $v$ to determine the outgoing edge (specified by a port number) along which the packet should be sent. In the *designer port model*, port numbers are assigned during preprocessing, whereas in the *fixed port model*, port numbers are assigned adversarially. This forwarding process continues until the packet reaches its destination.

A routing scheme has *stretch t* if, for every source–destination pair, the path length taken by the scheme is at most $t$ times the length of a shortest path in the network.

In this paper, we consider the underlying network to be a metric space. The algorithm first chooses an *overlay network* on which routing is performed. The main objective is to minimize the size of the routing tables stored at each vertex.

**Ackermann functions.** Following standard notions (e.g., [NS07, AS24]), we introduce the follwing Ackermann function.

**Definition 2.1** (Ackermann functions)**.** For all $k \ge 0$, the functions $A(k,n)$ and $B(k,n)$ are defined as follows:

$$A(0,n) := 2n, \quad \text{for all } n \ge 0,$$

$$A(k,n) := \begin{cases} 1 & \text{if } k \ge 1 \text{ and } n = 0, \\ A(k-1, A(k, n-1)) & \text{if } k \ge 1 \text{ and } n \ge 1, \end{cases}$$

$$B(0,n) := n^2, \quad \text{for all } n \ge 0,$$

$$B(k,n) := \begin{cases} 2 & \text{if } k \ge 1 \text{ and } n = 0, \\ B(k-1, B(k, n-1)) & \text{if } k \ge 1 \text{ and } n \ge 1. \end{cases}$$

Then, we have the following definition of the inverse Ackermann function:

**Definition 2.2** (Inverse Ackermann function)**.** For all $k \geq 0$, the function $\alpha_k(n)$ is defined as follows:

$$\alpha_{2k}(n) := \min\{\, s \geq 0 : A(k,s) \geq n \,\}, \quad \text{for all } n \geq 0,$$
$$\alpha_{2k+1}(n) := \min\{\, s \geq 0 : B(k,s) \geq n \,\}, \quad \text{for all } n \geq 0.$$

One can easily see that $\alpha_0(n) = \lceil n/2 \rceil$, $\alpha_1(n) = \lceil \sqrt{n} \rceil$, $\alpha_2(n) = \lceil \log n \rceil$, $\alpha_3(n) = \lceil \log\log n \rceil$, $\alpha_4(n) = \log^* n$, $\alpha_5(n) = \lfloor \frac{1}{2} \log^* n \rfloor$, etc.

## 3 Bounded treewidth tree covers with small hop-diameter

In this section, we show tight tradeoff between treewidth and hop-diameter for 1-spanners of tree metrics. In particular, the upper bound (Theorem 1.1) is proved in Section 3.1 and the matching lower bound (Theorem 1.2) is proved in Section 3.2.

The following two claims are used in the proofs of both theorems.

*Claim* 3.1. There is an absolute constant $\gamma$ such that for $\alpha \in \{0,1\}$, every integer $k \geq 4$, every $x > 1$ where the expression is defined, it holds

$$\frac{2}{k} < x^{2/k} - \left( x - \left(\frac{k}{k-2}\right)^{(k-2)/2} \cdot x^{(k-2)/k} - \alpha \right)^{2/k} < \frac{\gamma}{k} \tag{1}$$

*Proof.* We rewrite the expression as follows.

$$x^{2/k} \left( 1 - \left( 1 - \left(\frac{k}{k-2}\right)^{(k-2)/2} \cdot x^{-2/k} - \alpha x^{-1} \right)^{2/k} \right) \tag{2}$$

Using Maclaurin expansion, we have that $(1+y)^{2/k} = 1 + \frac{2}{k}y + \frac{2-k}{k^2} \cdot (1+\zeta)^{\frac{2}{k}-2} \cdot y^2$, where $\zeta$ is a number between 0 and $y$. We set $y = -\left(\frac{k}{k-2}\right)^{(k-2)/2} \cdot x^{-2/k} - \alpha x^{-1}$.

$$\left( 1 - \left(\frac{k}{k-2}\right)^{(k-2)/2} \cdot x^{-2/k} - \alpha x^{-1} \right)^{2/k} = \tag{3}$$

$$1 - \frac{2}{k}\left( \left(\frac{k}{k-2}\right)^{(k-2)/2} x^{-2/k} + \alpha x^{-1} \right) - \frac{k-2}{k^2}(1+\zeta)^{\frac{2}{k}-2} \left( \left(\frac{k}{k-2}\right)^{(k-2)/2} \cdot x^{-2/k} + \alpha x^{-1} \right)^2 \tag{4}$$

Plugging Equation (3) into Equation (2), we obtain the following.

$$x^{2/k} \left( 1 - \left( 1 - \left(\frac{k}{k-2}\right)^{(k-2)/2} \cdot x^{-2/k} - \alpha x^{-1} \right)^{2/k} \right) =$$
$$\frac{2}{k}\left( \left(\frac{k}{k-2}\right)^{(k-2)/2} + \alpha x^{-(k-2)/k} \right) + \frac{k-2}{k^2}(1+\zeta)^{\frac{2}{k}-2} \left( \left(\frac{k}{k-2}\right)^{(k-2)/2} \cdot x^{-1/k} + \alpha x^{-(k-1)/k} \right)^2$$

The lower bound in Equation (1) holds because $-1 < y < \zeta < 0$ and $x > 1$. Next we prove the upper bound.

$$\frac{2}{k}\left( \left(\frac{k}{k-2}\right)^{(k-2)/2} + \alpha x^{-(k-2)/k} \right) + \frac{k-2}{k^2}(1+\zeta)^{\frac{2}{k}-2} \left( \left(\frac{k}{k-2}\right)^{(k-2)/2} \cdot x^{-1/k} + \alpha x^{-(k-1)/k} \right)^2 <$$
$$\frac{2(e + x^{-(k-2)/k}) + (ex^{-1/k} + x^{-(k-1)/k})^2}{k}$$

The right-hand side is decreasing with $x$ in the whole domain and we can upper bound it by taking $x = 1$.

$$\frac{2(e + x^{-(k-2)/k}) + (ex^{-1/k} + x^{-(k-1)/k})^2}{k} < \frac{(e+1)(e+3)}{k}$$

Letting $\gamma = (e + 1)(e + 3)$, the upper bound from Equation (1) follows. □

*Claim* 3.2. For every $3 \le k \le \frac{2}{\ln(2e)} \ln \log n$, it holds that $(\frac{k}{k-2})^{(k-2)/2} (\log n)^{(k-2)/k} \le (\log n)/2$.

*Proof.* We have that $k \le \frac{2}{\ln(2e)} \ln \log n$. Rearranging the last inequality, we have that $e(\log n)^{(k-2)/k} \le (\log n)/2$. The proof is completed by observing that $(\frac{k}{k-2})^{(k-2)/2}$ is monotonically increasing for $k \ge 3$ and $\lim_{k\to\infty}(\frac{k}{k-2})^{(k-2)/2} = e$. □

*Claim* 3.3. For every $x \ge 1$ and $k \ge 4$, $x^{\sqrt{(k-2)/k}} \le x - \frac{x \ln x}{k} + \frac{x \ln^2 x}{k^2}$

*Proof.* Let $p := 1/k$. Then, $x^{\sqrt{(k-2)/k}} = x^{\sqrt{1-2p}}$. Using Taylor series around $p = 0$, we have that $x^{\sqrt{1-2p}} = x - px \ln x + R(p)$. The term $R(p)$ has the following form for $0 < \xi < p$.

$$R(p) = \frac{p^2}{2} \cdot x^{\sqrt{1-2\xi}} \ln x \left( \frac{\ln x}{1 - 2\xi} - (1 - 2\xi)^{-3/2} \right) \le \frac{p^2}{2} \cdot x^{\sqrt{1-2\xi}} \ln x \cdot \frac{\ln x}{1 - 2\xi} \le p^2 x \ln^2 x$$

To finish the proof, we replace $p$ by $1/k$. □

### 3.1 Upper bound

In this section we prove Theorem 1.1.

**Theorem 1.1.** *For every $n \ge 1$ and every $k = O(\log \log n)$, every $n$-vertex tree admits a shortcutting with hop-diameter $k$ and treewidth $O(k \log^{2/k} n)$ for even $k$ and $O(k(\frac{\log n}{\log \log n})^{2/(k-1)})$ for odd $k \ge 3$.*

The following lemma will be used in proving the theorem.

**Lemma 3.1** (Cf. Lemma 1 in [FL22b]). *Given a parameter $\ell \in \mathbb{N}$ and an $n$-vertex tree $T$, there is a set $X$ of at most $\frac{2n}{\ell+1} - 1$ vertices such that every connected component $C$ of $T \setminus X$ is of size at most $\ell$ and has at most 2 outgoing edges towards $X$. Furthermore, if $C$ has outgoing edges towards $x, y \in X$, then necessarily $x$ is an ancestor of $y$, or vice versa.*

#### 3.1.1 Hop-diameter 2

**Lemma 3.2.** *For every tree metric $M_T = (T, \delta_T)$ induced by a tree $T$ there is a 1-spanner $H_2$ with hop-diameter 2 and treewidth $O(\log n)$.*

*Proof.* Consider the following recursive construction due to [Sol13, AS24, NS07] which produces a set of edges $E_2$ of $H_2$. Take a centroid vertex $v$ of $T$ and add an edge between $v$ and every other vertex of $T$ to $E_2$. Recurse on each subtree of $T \setminus v$. Stop whenever $T$ is a singleton.

Let $\mathcal{T}_1, \dots, \mathcal{T}_g$ be the tree decompositions of spanners constructed for the subtrees in $T \setminus v$. Create a new bag $B$ containing only the vertex $v$ and add $v$ to every bag in each $\mathcal{T}_1, \dots, \mathcal{T}_g$. The tree decomposition of $H_2$ is obtained by connecting $B$ to the roots of each of $\mathcal{T}_1, \dots, \mathcal{T}_g$. The treewidth satisfies recurrence $W_2(1) = 0$ and $W_2(n) = W_2(n/2) + 1$, which has solution $W_2(n) = O(\log n)$.

Consider any two vertices $u, v \in T$. Let $w$ be the centroid vertex used in the last recursive call where $u$ and $v$ belonged to the same tree. By construction, $H_2$ contains edges $(w, u)$ and $(w, v)$, meaning that there is a 2-hop path between $u$ and $v$. Vertex $w$ is contained on the path between $u$ and $v$ in $T$, meaning that the stretch of this path is 1. □

### 3.1.2 Hop-diameter 3

**Lemma 3.3.** *For every tree metric $M_T = (T, \delta_T)$ induced by a tree $T$ there is a 1-spanner $H_3$ with hop-diameter 3 and treewidth $O(\log n / \log\log n)$.*

*Proof.* Let $\ell_3 = \log n / \log\log n$. This value will not change across different recursive levels when the subtree sizes shrink.

Consider the following recursive construction for constructing the edge set of $E_3$ of $H_3$. Let $X$ be a set of vertices for $T$ as in Lemma 3.1 with parameter $\ell = n/\ell_3$ so that $|X| = O(\ell_3)$. Connect the vertices of $X$ by a clique and add those edges to $E_3$. Next, do the following for every subtree $T'$ in $T \setminus X$. Let $u$ and $v$ be two vertices from $X$ to which $T'$ has outgoing edges. Connect $u$ and $v$ to every vertex in $T'$ and add the edges to $E_3$. Proceed recursively with $T'$. The base case occurs whenever the size of the tree is at most $\ell_3$. In the base case, we connect the vertices by a clique.

Let $\mathcal{T}_1, \ldots, \mathcal{T}_g$ be the tree decompositions of spanners constructed for the trees in $T \setminus X$. For every $\mathcal{T}_i$, let $u$ and $v$ be the two vertices from $X$ adjacent to the corresponding subtree $W_i$ in $T \setminus X$. (Lemma 3.1 guarantees that there is at most two such vertices; it is possible that $u = v$.) Add $u$ and $v$ to each bag in $\mathcal{T}_i$. Construct a new bag $B$ containing all the vertices in $X$ and connect it to the roots of each $\mathcal{T}_1, \ldots, \mathcal{T}_g$. The treewidth of $H_3$ satisfies $W_3(n) \le n$ for $n \le \ell_3$ and $W_3(n) \le W_3(n/\ell_3) + 2$ for $n > \ell_3$. Recall that $\ell_3$ is fixed and does not change across different levels of recursion. The recurrence satisfies $W_3(n) = O(\log n / \log\log n)$.

Consider any two vertices $u, v \in T$. Let $X$ be the set of vertices in the last recursive call when $u$ and $v$ were in the same tree. If $u$ and $v$ are considered in the same base case, then there is a direct edge between them. Otherwise, let $W_u$ (resp., $W_v$) be the subtree in $T \setminus X$ and let $u'$ (resp., $v'$) be the vertex in $X$ that is incident on $W_u$ (resp., $W_v$). By construction, $H_3$ contains edges $(u, u')$, $(v, v')$, and $(u', v')$. (The cases where $u' = u$ or $v' = v$ are handled similarly.) □

### 3.1.3 Hop-diameter $k \ge 4$

**Lemma 3.4.** *For every tree metric $M_T = (T, \delta_T)$ induced by a tree $T$ and every $k \ge 4$ there is a 1-spanner $H_k$ with hop-diameter $k$ and treewidth $O(k \log^{2/k} n)$ for even values of $k$ and $O(k(\frac{\log n}{\log\log n})^{2/(k-1)})$ for odd values of $k$.*

*Proof.* Let $\ell_k$ be such that $\log \ell_k = (\frac{k}{k-2})^{(k-2)/2} (\log n)^{(k-2)/k}$ for even values of $k$ and $\log \ell_k = (\frac{k}{k-2})^{(k-2)/2} (\log n / \log\log n)^{(k-2)/k}$ for odd values. By Claim 3.2, we have that $\ell_k \le \sqrt{n}$.

Consider the following recursive construction for constructing the edge set of $E_k$ of $H_k$. Let $X$ be a set of vertices for $T$ as in Lemma 3.1 such that $|X| = O(\ell_k)$ and every component of $T \setminus X$ has size at most $n/\ell_k$. Do the following for every subtree $T'$ of $T \setminus X$. Let $u$ and $v$ be two vertices from $X$ to which $T'$ has outgoing edges. Connect $u$ and $v$ to every vertex in $T'$ and add the edges to $E_k$. Proceed recursively with $T'$. The base case occurs whenever the size of the tree is at most $\ell_k$. In the base case, we connect the vertices of the considered tree using the construction with hop-diameter $k - 2$. To interconnect the vertices in $X$, we construct an auxiliary tree $T_X$; use the recursive construction with parameter $k - 2$ on $T_X$ and add all these edges to $H_k$. This concludes the description of $H_k$.

Next, we analyze the treewidth of $H_k$. Let $\mathcal{T}_1, \ldots, \mathcal{T}_g$ be the tree decompositions of spanners constructed for the trees in $T \setminus X$. For every $\mathcal{T}_i$, let $u$ and $v$ be the two vertices from $X$ adjacent to the corresponding subtree $W_i$ in $T \setminus X$. (Lemma 3.1 guarantees that there is at most two such vertices; it is possible that $u = v$.) Add $u$ and $v$ to each bag in $\mathcal{T}_i$. Let $\mathcal{T}_X$ be the tree decomposition of $T_X$, where $T_X$ is defined as in the previous paragraph. Since $\mathcal{T}_X$ is a valid tree decomposition and $T_X$ contains an edge $(u, v)$, then $\mathcal{T}_X$ contains a bag $L_{u,v}$

with both $u$ and $v$. Connect the root of $\mathcal{T}'$ to $L_{u,v}$. This concludes the description of the tree decomposition $\mathcal{T}$ of $T$. The treewidth of $\mathcal{T}$ satisfies the recurrence $W_k(n) = W_{k-2}(n)$ for $n \le \ell_k$ and $W_k(n) \le \max(W_{k-2}(\ell_k), W_k(n/\ell_k) + 2)$ otherwise.

We show that the recurrence satisfies $W_k(n) \le k \log^{2/k} n$ for even values of $k$. (The proof for odd values is similar.) For the base case we use $n \le \ell_k$, where $W_k(n) \le W_{k-2}(n)$. For the induction step, we assume that the hypothesis holds for all the values smaller than $n$. First, note that $W_{k-2}(\ell_k) \le (k-2)(\log \ell_k)^{2/(k-2)} = k(\log n)^{2/k}$.

$$\begin{aligned} W_k(n) &\le \max(k(\log n)^{2/k}, W_k(n/\ell_k) + 2) \\ &\le \max(k(\log n)^{2/k}, k(\log n - \log \ell_k)^{2/k} + 2) \\ &\le \max\left(k(\log n)^{2/k}, k\left(\log n - \left(\frac{k}{k-2}\right)^{(k-2)/2} \cdot (\log n)^{(k-2)/k}\right)^{2/k} + 2\right) \\ &\le \max\left(k(\log n)^{2/k}, k(\log n)^{2/k}\right) \end{aligned}$$

The last inequality follows from the left-hand side part of Equation (1) by setting $x = \log n$ and $\alpha = 0$.

To argue that $H_k$ is a 1-spanner with hop-diameter $k$, consider the last step of the construction where both $u$ and $v$ were in the same tree $T$. If $u$ and $v$ were considered in the same base case, then $T$ is equipped with a recursive construction with parameter $k-2$. By induction, there is a 1-spanner path between them with hop-diameter $k-2$. Otherwise, let $W_u$ (resp., $W_v$) be the subtree in $T \setminus X$ and let $u'$ (resp., $v'$) be the vertex in $X$ that is incident on $W_u$ (resp., $W_v$). By construction, there is a 1-spanner path between $u'$ and $v'$ with at most $k-2$ hops. (The cases where $u' = u$ or $v' = v$ are handled similarly.) □

## 3.2 Lower bound

We show the treewidth lower bound for 1-spanner of the uniform line metric $L_n = \{1, 2, \ldots, n\}$ with hop-diameter $k$. Due to the inductive nature of our argument, we prove a stronger version of the statement, which considers 1-spanners that could potentially use points outside of the given line metric.

**Theorem 1.2.** *For every $n \ge 1$, every shortcutting with hop-diameter $k$ for an $n$-point path must have treewidth:*

- *$\Omega(k \log^{2/k} n)$ for even $k$ and $\Omega(k(\frac{\log n}{\log\log n})^{2/(k-1)})$ for odd $k \ge 3$, whenever $k \le \frac{2}{\ln(2e)} \ln \log n$;*
- *$\Omega((\log\log n)^2/k)$ whenever $k > \frac{2}{\ln(2e)} \ln \log n$.*

We do so by arguing that any 1-spanner of $L_n$ has a large minor. It is well known that a graph with $K_t$ as a minor has treewidth at least $t-1$.

We prove lower bounds for even $k$; the lower bound for odd $k$ can be shown using the same argument.

### 3.2.1 Hop-diameter 2

**Lemma 3.5.** *For every $n \ge 1$ and every $n$-vertex line metric $L_n$, every 1-spanner with hop-diameter 2 has $K_{\lfloor \log n \rfloor + 1}$ as a minor.*

*Proof.* We prove the claim by complete induction over $n$. For the base case, we use $n = 1$, where the claim holds vacuously.

Let $H$ be a 1-spanner for $L_n$ with hop-diameter 2. Split $L_n$ into two consecutive parts, $L_1$ and $L_2$, of sizes $\lfloor n/2 \rfloor$ and $\lceil n/2 \rceil$, respectively. Let $H_1$ and $H_2$ be subgraphs of $H$ induced on $L_1$ and $L_2$, respectively. From the induction hypothesis, $H_1$ and $H_2$ have $K_{\log \lfloor n/2 \rfloor}$ and $K_{\log \lceil n/2 \rceil}$ as minors, respectively.

Consider the case where every point of $L_1$ has an edge in $H$ to some point in $L_2$. Then $K_{\log \lfloor n/2 \rfloor} \cup \{H_2\}$ induces a clique minor of size $\log \lfloor n/2 \rfloor + 1$. Consider the complementary case where $L_1$ has a point $p$ that does not have an edge in $H$ to any point in $L_2$. Then, every point in $L_2$ has a neighbor in $L_1$ because $H$ has hop-diameter 2. Thus, $K_{\log \lceil n/2 \rceil} \cup \{H_1\}$ induces a clique minor of size $\log \lceil n/2 \rceil + 1$. Hence, the minor size satisfies recurrence $W_2(n) = W_2(\lfloor n/2 \rfloor) + 1$, with a base case $W_2(1) = 1$. The solution is given by $W_2(n) = \lfloor \log n \rfloor + 1$. □

### 3.2.2 Hop-diameter $k \geq 4$

We give a proof for even values of $k$ such that $4 \leq k \leq \frac{2}{\ln(2e)} \ln \log n$ in Lemma 3.6. The proof for odd values is analogous. The proof for $k > \frac{2}{\ln(2e)} \ln \log n$ is given in Lemma 3.7.

**Lemma 3.6.** *For every $n \geq 1$, every even $4 \leq k \leq \frac{2}{\ln(2e)} \ln \log n$, and every $n$-vertex line metric $L$, every 1-spanner with hop-diameter $k$ has treewidth at least $c_1 k \log^{2/k} n - 1$, where $c_1$ is an absolute constant.*

*Proof.* Let $\ell_k$ be such that $\log \ell_k = (\frac{k}{k-2})^{(k-2)/2} (\log n)^{(k-2)/k}$ for even values of $k$. (The proof for odd values is similar. There, we choose $\ell_k$ so that $\log \ell_k = (\frac{k}{k-2})^{(k-2)/2} (\log n / \log \log n)^{(k-2)/k}$.) By Claim 3.2, we have that $\ell_k \leq \sqrt{n}$, whenever $k \leq \frac{2}{\ln(2e)} \ln \log n$.

Split $L$ into consecutive sets of points $L_1, L_2, \ldots, L_{\ell_k}$ of size $\lfloor n/\ell_k \rfloor$ each and ignore the remaining points. Let $H$ be a 1-spanner with hop-diameter $k$ for $L_n$. Our goal is to show that the size of a clique minor of $H$ can be lower bounded by the following recurrence.

$$W_k(n) \geq \min(W_{k-2}(\ell_k), W_k(n/(2\ell_k)) + 1) \text{ and } W_k(1) = 1 \tag{5}$$

We prove the statement by complete induction over $n$ and $k$. For the base case, we take $n = 1$, and $W_k(1) = 1 > c_1 k \log^{2/k} n - 1$.

We say that a point in $L_i$ is *global* if it has an edge to a point outside $L_i$ and *non-global* otherwise. We say that $L_i$ is global if all of its points are global and *non-global* otherwise. We consider two complementary cases as follows.

**Case 1: Every $L_i$ is non-global.** For every $L_i$ and $L_j$ the path between a non-global point in $L_i$ and a non-global point in $L_j$ must have the first (resp., last) edge inside $L_i$ (resp., $L_j$). Let $H'$ be obtained from $H$ by contracting each $L_i$ into a single vertex. (Clearly, $H[L_i]$ is connected.) Let $L'$ be the line metric obtained from $L_n$ by contracting every $L_i$ into a single point. Then $H'$ is a $(k-2)$-hop spanner of $L'$ with stretch 1. Thus, $H'$ has a minor of size $W_{k-2}(\ell_k) \geq c_1(k-2) \log^{2/(k-2)} \ell_k - 1 = c_1 k \log^{2/k} n - 1$.

**Case 2: Some $L_i$ is global.** Let $\{L_l, L_r\} = L \setminus L_i$, so that $L_l$ (resp., $L_r$) is on the left (resp., right) of $L_i$. (Possibly $L_l = \emptyset$ or $L_r = \emptyset$.) At least $|L_i|/2$ points in $L_i$ have edges to either $L_l$ or $L_r$. Without loss of generality, we assume the former. Let $L_i' \subseteq L_i$ be the subset of points that have edges to $L_l$ and let $H_i$ be the subgraph of $H$ restricted to preserving distances in $L_i$. Inductively, $H_i$ has a clique minor of size at least $W_k(n/(2\ell_k))$. (Since $H_i$ is a 1-spanner, it does not include any point outside of $L_i$.) Then $H_i$ and $L_l$ are vertex-disjoint (because we are considering 1-spanners) and hence their union has a clique minor of size $W_k(n/(2\ell_k)) + 1$. Thus, $W_k(n)$ satisfies eq. (5), which we lower bound next.

$$
\begin{aligned}
W_k(n) &\geq \min(c_1 k(\log n)^{2/k} - 1, W_k(n/(2\ell_k)) + 1) \\
&\geq \min(c_1 k(\log n)^{2/k} - 1, c_1 k(\log n - \log \ell_k - 1)^{2/k}) \\
&\geq \min\left(c_1 k(\log n)^{2/k} - 1, c_1 k\left(\log n - \left(\frac{k}{k-2}\right)^{(k-2)/2} \cdot (\log n)^{(k-2)/k} - 1\right)^{2/k}\right) \\
&\geq \min\left(c_1 k(\log n)^{2/k} - 1, c_1 k(\log n)^{2/k} - 1\right)
\end{aligned}
$$

The last inequality follows from Equation (1) by replacing $x = \log n$ and $\alpha = 1$. □

**Lemma 3.7.** *For every $n \geq 1$, every $k > \frac{2}{\ln(2e)} \ln \log n$, every 1-spanner with hop-diameter $k$ for an $n$-vertex line metric has treewidth at least $c_2(\log \log n)^2/k$, for an absolute constant $c_2$.*

*Proof.* We set $\ell_k$ so that $\log \log \ell_k = \sqrt{\frac{k-2}{k}} \log \log n$. (We use $\log(\cdot) := \log_2(\cdot)$.) For the clarity of exposition, we ignore the rounding issues. We note that $1 \leq \ell_k \leq n$. Using the same argument as in Lemma 3.6, we have $W_k(n) \geq \min(W_{k-2}(\ell_k), W_k(n/(2\ell_k)) + 1)$. We prove the lemma by induction, where the base case is Lemma 3.6 whenever $k > \frac{2}{\ln(2e)} \ln \log n$. Our goal is to prove that $W_{k-2}(\ell_k) \geq c_2(\log \log n)^2/k$ and $W_k(n/(2\ell_k)) + 1 \geq c_2(\log \log n)^2/k$. For the first inequality we distinguish two cases. If $k - 2 \leq \frac{2}{\ln(2e)} \ln \log(\ell_k)$, then by Lemma 3.6 we have

$$
\begin{aligned}
W_{k-2}(\ell_k) &\geq c_1(k-2) \log^{\frac{2}{k-2}} \ell_k - 1 \geq c_1(k-2) \log^{\frac{\ln(2e)}{\ln \log \ell_k}} \ell_k - 1 = 2ec_1(k-2) - 1 \geq ec_1 k - 1 \\
&\geq ec_1 \cdot \left(\frac{2}{\ln(2e)}\right)^2 \frac{(\ln \log \ell_k)^2}{k} - 1 \geq c_2 \frac{(\log \log \ell_k)^2}{k-2}
\end{aligned}
$$

The penultimate inequality holds because $k \geq \frac{2}{\ln(2e)} \ln \log(n) \geq \frac{2}{\ln(2e)} \ln \log(\ell_k)$. The last inequality holds for a proper choice of constant $c_2$. If $k - 2 > \frac{2}{\ln(2e)} \ln \log(\ell_k)$, we have $W_{k-2}(\ell_k) \geq c_2 \frac{(\log \log \ell_k)^2}{k-2}$ by the induction hypothesis. Hence, in both cases, we have:

$$
W_{k-2}(\ell_k) \geq c_2 \cdot \frac{(\log \log \ell_k)^2}{k-2} = c_2 \cdot \frac{\left(\sqrt{\frac{k-2}{k}} \log \log n\right)^2}{k-2} = c_2 \cdot \frac{(\log \log n)^2}{k}
$$

For the second inequality, let $x = \log n$. We have $\log \ell_k = x^{\sqrt{(k-2)/k}}$. Since $\frac{n}{2\ell_k} < n$, we have that $k > \frac{2}{\ln(2e)} \ln \log \frac{n}{2\ell_k}$ and the induction hypothesis gives the following.

$$
W_k\left(\frac{n}{2\ell_k}\right) + 1 \geq \frac{c_2 \left(\log \log \left(\frac{n}{2\ell_k}\right)\right)^2}{k} + 1 = \frac{c_2 \log^2\left(x - x^{\sqrt{(k-2)/k}} - 1\right)}{k} + 1
$$

To show that the right-hand side is at least $c_2 \log^2(x)/k$, it suffices to show the following:

$$
\log^2\left(x - x^{\sqrt{(k-2)/k}} - 1\right) + \frac{k}{c_2} \geq \log^2 x \tag{6}
$$

From Claim 3.3 we have $x^{\sqrt{(k-2)/k}} \leq x - \frac{x \ln x}{k} + \frac{x \ln^2 x}{k^2}$.

$$
\begin{aligned}
x - x^{\sqrt{(k-2)/k}} - 1 &\geq x - \left(x - \frac{x \ln x}{k} + \frac{x \ln^2 x}{k^2}\right) - 1 \\
&= \frac{x \ln x}{k}\left(1 - \frac{\ln x}{k}\right) - 1 \\
&> \frac{x \ln x}{10k} - 1
\end{aligned}
$$

The last inequality holds because $k > \frac{2}{\ln(2e)} \ln(x)$. We next consider two cases. If $\frac{x \ln x}{10k} < 2$, then $\frac{k}{c_2} > \frac{x \ln x}{20 c_2} \geq \log^2 x$ and Equation (6) is proved. Otherwise, we proceed as follows.

$$
\begin{aligned}
\log^2\left(\frac{x \ln x}{10k} - 1\right) + \frac{k}{c_2} &\geq \log^2\left(\frac{x \ln x}{20k}\right) + \frac{k}{c_2} \\
&= \left(\log x + \log \frac{\ln x}{20k}\right)^2 + \frac{k}{c_2} \\
&= \log^2 x + 2(\log x) \log \frac{\ln x}{20k} + \log^2 \frac{\ln x}{20k} + \frac{k}{c_2} \\
&\geq \log^2 x
\end{aligned}
$$

The last inequality holds for any $c_2 \leq 1/10$. □

# 4 Bounded arboricity tree covers

Throughout this section, we use the following lemma.

**Lemma 4.1.** *If every edge of a graph $G = (V, E)$ can be oriented such that the maximum in-degree of every vertex is at most $d$, then the arboricity of $G$ is at most $d + 1$.*

## 4.1 Line metrics

In this section, we show a construction for line metrics (Theorem 1.4). We shall use a modification of the following well-known result.

**Theorem 4.1** (Cf. [AS24, BTS94, Sol13])**.** *For every $n \geq 2$ and $k \geq 2$, every $n$-point tree metric admits a 1-spanner with hop-diameter $k$ and $O(n\alpha_k(n))$ edges.*

We next state a slightly modified version of the previous theorem. The first statement concerns hop-diameter 1.

**Lemma 4.2.** *Let $n \geq 2$ be an arbitrary integer. Let $L$ be a line metric induced by a set of $n$ points on a line so that between every two points there is $n$ Steiner points. Let $S$ denote the set of Steiner points. Then, $L$ admits a Steiner 1-spanner with hop-diameter 2 such that the Steiner points belong to $S$ and every vertex in $L \cup S$ has a constant in-degree.*

*Proof.* Interconnect the vertices in $L$ by a clique. Consider an arbitrary clique edge $(u, v)$ and split it into two using a Steiner point $w$. Orient the edges $(u, w)$ and $(w, v)$ into $w$. By using a fresh Steiner point for every clique edge, we obtain the guarantees from the statement. □

Next, we state the general version.

**Lemma 4.3.** *Let $n \geq 2$ and $k \geq 2$ be two arbitrary integers. Let $L$ be a line metric induced by a set of $n$ points on a line so that between every two points there is $\alpha_k(n)$ Steiner points. Let $S$ denote the set of Steiner points. Then, $L$ admits a Steiner 1-spanner with hop-diameter $2k$ such that the Steiner points belong to $S$ and every vertex in $L \cup S$ has a constant in-degree.*

*Proof.* We prove the lemma by induction over $k$. For $k = 2$, we take a central vertex $c$ in $L$ and connect it to every other point in $L$; orient the edges outwards from $c$. Proceed recursively with the two halves. This way we obtain a 1-spanner for $L$ with hop-diameter 2. Denote by $E$ the edge set of this spanner. The depth of the recursion in the construction is $O(\log n)$, and the size of $S$ is $n\alpha_2(n) = n \log n$. Every edge in $E$ has in-degree 1 for every recursion level. We can split every such edge into two using a fresh Steiner point for each recursion level. This concludes the proof for $k = 2$.

Consider now an arbitrary $k$. Divide $L$ into intervals of size $\alpha_{k-2}(n)$ using $n/\alpha_{k-2}(n)$ cut vertices. Denote by $C$ the set of cut vertices and invoke the induction hypothesis on $C$ with parameter $k-2$. Let $E'$ be the obtained set of edges. Let $E_C$ be obtained by connecting every cut vertex to every point in the two neighboring intervals. Let this edge set be $E_C$. Proceed recursively with parameter $k$ on each of the intervals.

To analyze the in-degree, we observe that the depth of the recursion with parameter $k$ is $O(\alpha_k(n))$, which coincides with the number of Steiner vertices between every two points in $L$. One level of recursion contributes a constant in-degree to each vertex in the construction. This means that we can split all such vertices into two and use a fresh Steiner point at each recursion level. This concludes the proof. □

We restate Theorem 1.4 here for convenience.

**Theorem 1.4.** *For every $n \geq 1$ and every even $k \geq 2$, every $n$-point path admits a shortcutting with hop-diameter $k$ and arboricity $O(\alpha_{k/2+1}(n))$.*

*Proof.* Let $L_n$ be an arbitrary line metric. For an integer $k' \geq 2$, we describe a construction of a 1-spanner $H$ for $L_n$ with hop-diameter $4k'-2$ and arboricity $O(\alpha_{2k'}(n))$.

Consider a set of $n' = n/\alpha_{2k'-2}(n)$ equally-spaced *cut vertices* dividing $L_n$ into intervals of size $\alpha_{2k'-2}(n)$. To construct the 1-spanner $H$, we connect every cut vertex to all the vertices in its two neighboring intervals. Denote the corresponding edge by $E_C$. Let $C$ be the set of cut vertices. Let $E'$ be obtained by invoking Lemma 4.3 with parameter $2k'-2$ on $C$ using $L_n$ as Steiner points. Proceed recursively with each of the intervals.

To analyze the arboricity, we will show that the edges in $E_C$ and $E'$ can be oriented so that the in-degree of every vertex is constant. Orient every edge in $E_C$ so that it goes out of the corresponding cut vertex. Since every interval is adjacent to at most two cut vertices, the in-degree of every point with respect to $E_C$ is at most 2. By Lemma 4.3, the edges in $E'$ have a constant in-degree on $L_n$. In conclusion, $E_C$ and $E'$ contribute $O(1)$ to the in-degree of each vertex in $L_n$. The number of recursion levels $\ell(n)$ satisfies the recurrence $\ell(n) = \ell(\alpha_{2k'-2}(n)) + O(1)$, which has solution $\ell(n) = \alpha_{2k'}(n)$. Every recursion level contributes $O(1)$ to the in-degree of vertices, meaning that the overall in-degree in $H$ is $O(\alpha_{2k'}(n))$. The hop-diameter of $H$ is $2 + 2 \cdot (2k'-2) = 4k'-2$. This concludes the description of a 1-spanner with hop-diameter $4k'-2$ and arboricity $O(\alpha_{2k'}(n)$.

Note that we have only shown how to get a tradeoff of hop-diameter $4k'-2$ and arboricity $O(\alpha_{2k'}(n))$. We could similarly get a construction with hop-diameter $4k'$ and arboricity $O(\alpha_{2k'+1}(n))$ for a parameter $k' \geq 1$. Specifically, we divide $L_n$ into intervals of size $\alpha_{2k'-1}(n)$. The cut vertices are interconnected using Lemma 4.3 with hop-diameter $2k'-1$ (and Lemma 4.2 when $k' = 1$). The hop-diameter is $2 + 2(2k'-1) = 4k'$ and the arboricity is $O(\alpha_{2k'+1}(n))$ using a similar argument. □

The construction for ultrametric is a simple adaptation of the construction for line metrics, which we will show next.

## 4.2 Ultrametrics and doubling metrics

A metric $(X, d_X)$ is an *ultrametric* if it satisfies a strong form of the triangle inequality: for every $x, y, z$, $d_X(x, z) \leq \max\{d_X(x, y), d_X(y, z)\}$. It is well known that an ultrametric can be represented as a *hierarchical well-separated tree* (HST).

More precisely, an $(s, \Delta)$-*HST* is a tree $T$ where (i) each node $v$ is associated with a label $\Gamma_v$ such that $\Gamma_v \geq s \cdot \Gamma_u$ whenever $u$ is a child of $v$ and (iii) each internal node $v$ has at most $\Delta$ children. Parameter $s$ is called the *separation* while $\Delta$ is called the degree of the HST of the HST. Let $L$ be the set of leaves of $T$. The labels of internal nodes in $T$ induce a metric $(L, d_L)$ on the leaves, called leaf-metric, where for every two leaves $u, v \in L$, $d_L(u, v) = \Gamma_{\text{lca}(u,v)}$ where lca is the lowest common ancestor of $u$ and $v$. It is well-known, e.g., [BLMN04], that $(L, d_L)$ is an ultrametric, and that any ultrametric is isomorphic to the leaf-metric of an HST.

Chan and Gupta [CG06] showed that any $(1/\epsilon, 2)$-HST can be embedded into the line metric with (worst-case) distortion $1+O(\epsilon)$. Therefore, by applying Theorem 1.4, we obtain a $(1+O(\epsilon))$-spanner with hop diameter $k$ and arboricity $O(\alpha_{k/2+1}(n))$ for $(1/\epsilon, 2)$-HST. In our setting, we are interested in large-degree $(k, \Delta)$-HST where $s = 1/\epsilon$ and $\Delta = \text{poly}(1/\epsilon)$; the embedding result by Chan and Gupta [CG06] no longer holds for these HSTs. Instead, we directly apply our technique for the line metric to get a $(1 + O(\epsilon))$-spanner with low-hop diameter.

**Lemma 4.4.** *Let $\epsilon \in (0, 1), \Delta > 0$ be parameters, and $k$ be an even positive integer. Then, any $(1/\epsilon, \Delta)$-HST with $n$ leaves admits a $(1+O(\epsilon))$-spanner with hop-diameter $k$ and arboricity $O(\alpha_{k/2+1}(n))$.*

*Proof.* Let $T$ be the $(1/\epsilon, \Delta)$-HST and let $M_T$ be the metric induced by $T$. For an integer $k' \geq 2$, we describe a construction of a $(1 + O(\epsilon))$-spanner for $M_T$ with hop-diameter $4k' - 2$ and arboricity $O(\alpha_{2k'}(n))$. The construction is similar to that in Theorem 1.4.

Let $C$ be the set of internal nodes of $T$, called *cut verices*, such that the subtrees rooted at these nodes has size $\alpha_{2k'-2}(n)$. The number of cut vertices is $|C| \leq n/\alpha_{2k'-2}(n)$. First, connect every cut vertex to all of its descendants in $T$ and let the corresponding set of edges be $E_C$. Next, let $E$ be the set of edges interconnecting the cut vertices using Theorem 4.1 with hop-diameter $2k' - 2$ and $O(n'\alpha_{2k'-2}(n')) = O(n)$ edges. We construct a set $E'$ by subdividing every edge $(u, v) \in E$ into two edges using a vertex, say $x$, in the subtree rooted at $u$. The spanner $H$ is obtained using the edges in $E_C$ and those in $E'$. Finally, the recursive construction is applied to each subtree rooted at a vertex in $C$. This concludes the description of the recursive construction of $H$.

The same argument in Theorem 1.4 implies that the arboricity is $O(\alpha_{2k'}(n))$. The stretch is $(1+O(\epsilon))$ since the path $u \to x \to v$ between two cut vertices $u$ and $v$ has stretch $(1+O(\epsilon))$. □

To construct a low-hop spanner with small arboricity for doubling metrics (Theorem 1.5), we will rely on the ultrametric cover by Filtser and Le [FL22a]. Following their notation, for a given metric space $(X, d_X)$, we say that a collection $\mathcal{T}$ of at most $\tau$ different $(s, \Delta)$-HSTs such that (i) for every HST $T \in \mathcal{T}$, points in $X$ are leaves in $T$, and (ii) for every two points $x, y \in X$, $d_X(x, y) \leq d_T(x, y)$ for every $T \in \mathcal{T}$, and there exists a tree $T_{xy} \in \mathcal{T}$ such that $d_X(x, y) \leq \rho \cdot d_{T_{xy}}(x, y)$.

**Theorem 4.2** (Cf. Theorem 3.4 in [FL22a])**.** *For every $\epsilon \in (0, 1/6)$, every metric with doubling dimension $d$ admits an $(\epsilon^{-O(d)}, 1 + O(\epsilon), 1/\epsilon, \epsilon^{-O(d)})$-ultrametric cover.*

**Theorem 1.5.** *Let $k$ be an even integer and let $\epsilon \in (0, 1/6)$ be an arbitrary parameter. Then, for every positive integer $n$, every $n$-point metric with doubling dimension $d$ admits a $(1+\epsilon)$-spanner with hop-diameter $k$ and arboricity $\epsilon^{-O(d)}\alpha_{k/2+1}(n)$.*

*Proof.* Let $\mathcal{T}$ be the $(\epsilon^{-O(d)}, 1+O(\epsilon), 1/\epsilon, \epsilon^{-O(d)})$-ultrametric cover in Theorem 4.2 for the input doubling metric. The theorem then follows by applying Lemma 4.4 to each of $(1/\epsilon, \epsilon^{-O(d)})$-HST in $\mathcal{T}$ and taking the union of the resulting spanners. □

### 4.3 General tree metrics

**Theorem 1.3.** *For every two integers $n \geq 1$ and $k \geq 1$ and every $n$-vertex tree $T$, there is a shortcutting with hop-diameter $k$ and arboricity $O(\log^{12/(k+4)} n)$. Moreover, when the height of the tree is $h$, then the arboricity is $O(h^{6/(k+4)})$.*

*Proof of Theorem 1.3 for height $h$.* We first show how to prove the theorem for a tree $T$ with height bounded by $h$. This construction gives the main ideas used for general trees. Let $k'$ be an arbitrary integer. We show a recursive construction with hop-diameter $2k'$. In particular, we show how to shortcut $T$ so that between every ancestor and descendant it is possible to go using $k'$ hops. The construction is the same as in Lemma 3.2: take the root of the tree, connect it to every descendant and proceed recursively with each of its children. By orienting the edges from the roots to descendants, we have that the in-degree of every vertex is bounded by $h$. Let $A_1(h)$ denote the obtained in-degree (and thus arboricity) of the shortcutting. We have that $A_1(h) = h$.

We next show the bound for an arbitrary $k' = 1 + 3g$ for an integer $g \geq 1$. Let $\ell$ be a parameter to be set later. Consider the tree levels so that the root is at level 0. Designate as the *cut vertices* all the vertices at levels $\ell, 2\ell, 3\ell \ldots$. Denote the set of cut vertices by $C$. Consider the set $S$, consisting of all the parents of vertices in $C$. Let $E_{CS}$ be obtained by interconnecting all the vertices in $C$ to their parents. Each such edge is oriented from a vertex in $S$ towards the vertex in $C$. Next, connect every vertex in $S$ to its first $\ell - 1$ ancestors until the occurrence of the first cut vertex; let the corresponding edge set be $E_S$. Every such edge is oriented from the ancestors towards vertices in $S$. Let $c \in C$ be an arbitrary vertex at level $d$. Connect $c$ to all of its descendants at levels $d+1, d+2, \ldots, d+\ell-2$, i.e., until the nest cut vertex, and orient these edges from $c$ towards the descendants. The edge set $E_C$ is obtained by doing this for every vertex $c$ in $C$. Use a recursive construction with parameter $k' - 3$ on the subtree of $T$ induced by vertices in $C$. Finally, consider all the subtrees obtained by removing $C$ and $S$ from $T$ and apply the recursive construction with parameter $k'$ on each of the subtrees.

We next analyze the hop-diameter of the ancestor-descendant paths in this construction. Let $u$ and $v$ be two arbitrary vertices such that $v$ is an ancestor of $u$. The path between $u$ and $v$ in the shortcutting is as follows. By construction, $E_C$ contains an edge between $u$ and its ancestor cut vertex $c_u \in C$. Let $d_u$ be the highest cut vertex that is ancestor of $c_u$ and a descendant of $v$. There is a path consisting of at most $k' - 3$ hops between $c_u$ and $d_u$. Let $s_u \in S$ be the parent of $d_u$. The edge $(d_u, s_u)$ is in $E_{CS}$. Finally, $E_S$ contains an edge $(s_u, v)$. Clearly, the path consists of $k'$ hops.

We next analyze the in-degree of vertices in $T$. Let $A_{k'}(h)$ denote the in-degree of the construction with parameter $k'$. Then, $A_{k'}(h) = \ell - 1$ for the vertices in $S$, due to the orientation of the edges in $E_S$. For the vertices in $C$, we have $A_{k'}(h) = 1 + A_{k'-3}(h/\ell)$, because the edges in $E_{CS}$ add one to in-degree of every vertex and the dominant term is due to the recursive call with parameter $k' - 3$. Finally, all the other vertices have $A_{k'}(h) = 1 + A_k(\ell - 2)$, where the edges in $E_C$ contribute one to the in-degree and $A_k(\ell)$ is due to the recursive construction with parameter $k'$. Putting everything together, we have the following recurrence .

$$A_{k'}(h) = \max(\ell - 1, 1 + A_{k'-3}(h/\ell), 1 + A_{k'}(\ell - 2)) \tag{7}$$

We proceed to show inductively that for every $k' = 1 + 3g$ we have $A_{k'}(h) \leq h^{1/(g+1)}$. Since we have that $A_{k'}(h) \leq h$ for every $k' \geq 1$, we can disregard the third term in Equation (7). Thus, we obtain the following simplified recurrence: $A_{k'}(h) = \max(\ell, A_{k'-3}(h/\ell))$. Notice that we have replaced $\ell - 1$ by $\ell$ in the first term since it does not affect the solution asymptotically. We proceed to solve the recurrence.

$$\begin{aligned}
A_{k'}(h) &\leq \max(\ell, A_{k'-3}(h/\ell)) \\
&\leq \max(\ell, (h/\ell)^{1/g}) && \text{induction hypothesis} \\
&\leq \max(h^{1/(g+1)}, (h^{1-1/(g+1)})^{1/g}) && \text{setting } \ell = h^{1/(g+1)} \\
&\leq \max(h^{1/(g+1)}, h^{1/(g+1)}) \\
&= h^{1/(g+1)}
\end{aligned}$$

Thus $A_{k'}(h) \leq h^{1/(g+1)}$. In particular, we have that $g = (k-1)/3$ so the tradeoff is $k'$ versus arboricity $h^{3/(k'+2)}$. To get the tradeoff guaranteed in the statement, we observe that the hop-diameter is $2k'$.

□

*Proof of Theorem 1.3 for general trees.* We consider a heavy-light decomposition of $T$, constructed as follows. Start from the root $r$ down the tree each time following the child with the largest subtree. The obtained path is called the heavy path rooted at $r$. Continue recursively with every child of the vertices of the heavy path. Let $T'$ be obtained by contracting every heavy path in $T$ into a single vertex. It is well-known that the height of $T'$ is $\log n$, where $n$ is the number of vertices in $T$.

We start by employing the shortcutting procedure for bounded height trees on $T'$ with parameter $k'$ and explain how to adapt it to $T$. Consider an arbitrary edge $(u, v)$ in the shortcutting of $T$ such that $u$ is a descendant of $v$. Let $P_u$ (resp., $P_v$) be the heavy path in $T$ corresponding to $u$ (resp., $v$). Add an edge between the root $r_u$ of $P_u$ and its lowest ancestor on $P_v$. We do this for all the edges in the shortcutting of $T'$. For every heavy path $P$ in $T$, use the 4-hop construction with arboricity $O(\log\log n)$ from Theorem 1.4. In addition, connect via a direct edge every vertex in $p$ to the root of $P$. This concludes the description of the shortcutting for $T$.

Next, we analyze the hop-diameter of the obtained construction. Let $u$ and $v$ be two arbitrary vertices in $T$ and let $P_u$ and $P_v$ be the corresponding heavy paths in $T$. Denote by $p_u$ and $p_v$ the vertices in $T'$ corresponding to $P_u$ and $P_v$. Let $p_w$ be the LCA of $p_u$ and $p_v$ in $T'$. From the construction we know that there is a $k'$-hop path between $p_u$ and $p_w$ and between $p_v$ and $p_w$. Let $(p_v, p_a)$ be the first edge on the path from $p_u$ to $p_w$. The corresponding path in $T$ goes from $u$ to the parent of $P_u$ and from the root of $P_u$ to its lowest ancestor in $P_a$. We can similarly replace every edge on the path between $P_u$ and $P_w$ in $T'$ by two edges in $T$. We handle analogously the path between $P_v$ and $P_w$. The corresponding paths in $T$ go from $u$ to its lowest ancestor on $P_w$ and from $v$ to its lowest ancestor on $P_w$. Using the edges from the 4-hop construction on $P_w$, we join the two paths. The overall number of hops is $4k' + 4$. In particular, we achieve a hop-diameter of $4k' + 4$ with arboricity of $O(\log^{3/(k'+2)} n)$. Letting $k = 4k' + 4$, the arboricity is $O(\log^{12/(k+4)} n)$. □

## 5 Routing schemes

In this section, we show the results for routing in tree metrics and doubling metrics.

### 5.1 Routing in tree metrics

**Theorem 1.8.** *For every $n$ and every $n$-vertex tree $T$, there is a 3-hop routing scheme in the fixed-port model for the metric $M_T$ induced by $T$ with stretch 1 that uses $O(\log^2 n/\log\log n)$ bits per vertex.*

*Proof.* Our routing scheme is constructed on top of a 1-spanner $H_3$ of $M_T$ as described in Lemma 3.3. For a vertex $u \in T$, let $\mathrm{table}(u)$ denote its routing table and $\mathrm{label}(u)$ its label. First, assign a unique identifier $ID(u) \in \{1,\ldots,n\}$ to every vertex $u$ in $T$ and add it to $\mathrm{table}(u)$ and $\mathrm{label}(u)$. Equip the routing table and a label of every vertex $u \in T$ with an ancestor label $\mathrm{anc}(u)$ as in [AAK+06]. This adds $O(\log n)$ bits of memory per vertex. Using the ancestor labeling scheme from [AAK+06], we can determine, given two vertices $u$ and $v$, whether they are in ancestor-descendant relationship, and if so, whether $u$ is the ancestor of $v$ or the vice-versa.

Recall the recursive construction of $H_3$ with parameter $T$ as an input. Assign to each recursive call a unique integer $r_T$. Let $X$ be a set of vertices for $T$ as in Lemma 3.1 so that $|X| = \log n/\log\log n$. (This parameter is fixed and does not change across different recursive calls.) The vertices of $X$ are interconnected by a clique in $H_3$. For every vertex in $u \in X$, add to $\mathrm{table}(u)$ the information consisting of $C(u) = \langle r_T, \{\langle ID(v), \mathrm{port}(u,v), \mathrm{anc}(v)\rangle \mid v \in X \setminus \{x\}\}\rangle$. The memory occupied per every vertex in $X$ is $O(\log^2 n/\log\log n)$. (Note that the construction of $H_3$ guarantees that every vertex belongs to such a clique exactly once across all the recursive calls, meaning that $\mathrm{table}(u)$ contains only one such $C(u)$.) Let $T'$ be a subtree in $T \setminus X$. Let $u$ and $v$ be two vertices from $X$ to which $T'$ has outgoing edges. For every vertex $x \in T'$, add to $\mathrm{table}(x)$ the following: $\langle r_T, ID(u), \mathrm{port}(x,u)\rangle$ and $\langle r_T, ID(v), \mathrm{port}(x,v)\rangle$. Similarly, add to $\mathrm{label}(x)$ the following: $\langle r_T, ID(u), \mathrm{port}(u,x)\rangle$ and $\langle r_T, ID(v), \mathrm{port}(v,x)\rangle$. This information takes $(\log n)$ bits per recursive call $r_T$. The construction proceeds recursively with $T'$; the number of recursive calls every vertex participates in is at most $O(\log n/\log\log n)$.

Next we describe the routing algorithm. Let $u$ be the source and $v$ be the destination. First, check if $C(u)$ contains routing information leading directly to $v$. In this case, the algorithm outputs $\mathrm{port}(u,v)$ and the routing is complete. (This case happens when $u$ and $v$ are in the same clique during the construction.) Otherwise, go over $\mathrm{table}(u)$ and $\mathrm{label}(v)$ and find the last recursive call $r_T$ which is common to both $u$ and $v$. Next, consider $\mathrm{label}(v)$ and the two entries consisting $\langle r_T, ID(v_1), \mathrm{port}(v_1,v)\rangle$ and $\langle r_T, ID(v_2), \mathrm{port}(v_2,v)\rangle$, corresponding to $r_T$. If $v_1$ and $v_2$ are in $C(u)$, use $\mathrm{anc}(u)$, $\mathrm{anc}(v_1)$, $\mathrm{anc}(v_2)$, and $\mathrm{anc}(v)$ to decide whether to output $\mathrm{port}(u,v_1)$ or $\mathrm{port}(u,v_2)$. (This case happens when $u$, $v_1$, and $v_2$ are in $X$ in the recursive call $r_T$ and $v$ is not in it.) Finally, let $\langle r_T, ID(u_1), \mathrm{port}(u,u_1)\rangle$ and $\langle r_T, ID(u_2), \mathrm{port}(u,u_2)\rangle$ be the two entries corresponding to $r_T$ in $\mathrm{table}(u)$. Use $\mathrm{anc}(u)$ and $\mathrm{anc}(v)$ to decide whether to output $\mathrm{port}(u,u_1)$ or $\mathrm{port}(u,u_2)$. (This case happens when $u$ is not in $X$.) This concludes the description of the routing algorithm. □

### 5.2 Routing in doubling metrics

We next show how to extend the routing result in tree metrics to metrics with doubling dimension $d$. In particular, we prove the following theorem.

**Theorem 1.10.** *For every $n$ and every $n$-point metric with doubling dimension $d$, there is a 3-hop routing scheme with stretch $(1+\epsilon)$ that uses $O_{\epsilon,d}(\log^2 n/\log\log n)$ bits per vertex.*

Given a point set $P$ with doubling dimension $d$, we first construct a tree cover, using the tree cover theorem from [CCL+25].

**Theorem 5.1** ([CCL+25])**.** *Given a point set $P$ in a metric of constant doubling dimension $d$ and any parameter $\epsilon \in (0,1)$, there exists a tree cover with stretch $(1+\epsilon)$ and $\epsilon^{-\tilde{O}(d)}$ trees. Furthermore, every tree in the tree cover has maximum degree bounded by $\epsilon^{-O(d)}$.*

We use this specific tree cover theorem, since the authors also provide an algorithm for determining the "distance-preserving tree" given the labels of any two metric points.

**Lemma 5.1** ([CCL+25])**.** *Let $\epsilon \in (0,1)$. Let $T = \{T_1, \ldots, T_k\}$ be the tree cover for $P$ constructed by Theorem 5.1, where $k = \epsilon^{-\tilde{O}(d)}$. There is a way to assign $\epsilon^{-\tilde{O}(d)} \log n$-bit labels to each point in $P$ so that, given the labels of two vertices $x$ and $y$, we can identify an index $i$ such that tree $T_i$ is a "distance-approximating tree" for $u$ and $v$; that is, $\delta_{Ti}(x,y) \leq (1+\epsilon)\delta_P(x,y)$. This decoding can be done in $O(d \cdot \log(1/\epsilon))$ time.*

We equip each tree in the cover with the stretch-1 routing scheme from Theorem 1.8. This consumes overall $\epsilon^{-\tilde{O}(d)} \log^2 n / \log\log n$ bits per point in $P$. In addition, we add $\epsilon^{-\tilde{O}(d)} \log n$-bit labels to each point in $P$ as stated in Lemma 5.1. Given two points $x, y \in P$, we first employ the algorithm from Lemma 5.1 to find the tree in which the routing should proceed. Then, the routing proceeds on that specific tree using the routing algorithm from Theorem 1.8. This concludes the description of the compact routing scheme for doubling metrics.

# 6 Lower bound for routing in tree metrics

In this section, we prove the following theorem.

**Theorem 1.9.** *There is an infinite family of trees $T_n$, $n > 0$, such that any labeled fixed-port routing scheme with stretch 1 on a metric induced by $T_n$ has at least one vertex with total memory of $\Omega(\log^2 n / \log\log n)$ bits.*

Let $T$ be an unweighted tree and $M_T = (V, (V \times V), d_T)$ be a metric induced by $T = (V, E)$. The edges in $(V \times V) \setminus E$ are called *Steiner edges*. In this section we show that stretch-1 routing in tree metrics requires $\Omega(\log^2 n / \log\log n)$ bits per tree vertex.

**Hard instances.** We first describe the hard instances used in [FG02]. Let $t$, $h$, and $d$ be positive integers and let $\overline{T}_0$ be a complete $t$-ary rooted tree of height $h+1$. Let $T_0$ be a tree obtained by adding $d-t-1$ leaves at every internal vertex of $\overline{T}_0$ and $d-t$ leaves at the root. These added leaves are called *dummy leaves*. The number of vertices in $T_0$ is $n = (d-1) \cdot \frac{t^h - 1}{t-1} + 2$. Note that $T_0$ is uniquely defined by $t$, $h$, and $d$.

Let $T$ be a tree obtained from $\overline{T}_0$ as follows. Consider an internal vertex $u$ of $\overline{T}_0$ at height $i$, where the root has height $h$ and the leaves have height 0. Let $q_i = \frac{t^i - 1}{t-1}$. (The choice of $q_i$ coincides with the number of non-dummy vertices in a subtree rooted at any child of $u$.) Add $(d-t-1) \cdot q_i$ dummy leaves to $u$ if it is an internal node and $(d-t)q_i$ if it is the root. Note that both $T_0$ and $T$ are constructed based on $\overline{T}_0$ and there is a correspondence between non-dummy vertices of $T$ and the non-dummy vertices of $T_0$. In what follows, we shall use the same letter to denote some non-dummy vertex in $T$ and the corresponding non-dummy vertex in $T_0$.

*Claim* 6.1. The number of vertices in $T$ is $O(n \log^2 n)$.

*Proof.*

$$
\begin{aligned}
|T| &= |\overline{T}_0| + q_h + \sum_{i=1}^{h} t^{h-i} \cdot (d-t-1) \cdot q_i \\
&= \frac{t^{h+1}-1}{t-1} + \frac{t^h-1}{t-1} + \frac{d-t-1}{t-1} \cdot \sum_{i=1}^{h} t^{h-i} \cdot (t^i - 1) \\
&= \frac{t^{h+1}-1}{t-1} + \frac{t^h-1}{t-1} + \frac{(d-t-1)(ht^{h+1} - ht^h - t^h + 1)}{(t-1)^2}
\end{aligned}
$$

In [FG02], $d = t^h$ and $t = h = \lfloor (\log \sqrt{n}) / \log\log \sqrt{n} \rfloor$, so that $d = t^h \leq \sqrt{n}$. We proceed to upper bound $|T|$ as follows.

$$
|T| \leq 2t^{h+1} + dht^{h+1} \leq 2\sqrt{n}\log n + n\log^2 n = O(n\log^2 n)
$$

□

Using a reduction to the lower bound instances of [FG02], we will show that the memory requirement is $\Omega(\log^2 n' / \log\log n') = \Omega(\log^2 n / \log\log n)$.

**Reduction to relaxed routing.** Let $M_T$ be a metric induced by $T$. Similarly to [FG02], we consider a restricted problem of *relaxed routing* in $M_T$, where the destination vertex is a non-dummy vertex of $M_T$ and the source vertex is its ancestor. Our lower bound argument shows that relaxed routing in $M_T$ requires total memory of $\Omega(\log^2 n / \log\log n)$ bits per vertex. Since every routing scheme in an instance $M_T$ is also a relaxed routing scheme in the same instance, our lower bound applies to routing in $M_T$.

**Port numbering.** In [FG02], the authors consider a family $\mathcal{T}$ of instances where all the trees are isomorphic to $T_0$ and each instance correspond to a different port numbering of $T_0$. We consider a family of instances $\mathcal{T}'$ where every metric is isomorphic to $M_T$ and there is a *one-to-one correspondence* between instances in $\mathcal{T}$ and those in $\mathcal{T}'$. Consider an instance $\hat{T}_0$ from $\mathcal{T}$. We proceed to explain the port numbering in the corresponding instance $\hat{M}_T$ in $\mathcal{T}'$. Let $u$ be an internal vertex of $T$ at height $i$. Define the sets of edges $E_j$ as follows:

- For $1 \leq j \leq t$, let $E_j$ be the set of $q_i$ edges leading to the non-dummy descendants of $u$ in the subtree of $M_T$ rooted at the $j$th child of $u$.
- Partition the edges leading to dummy leaves adjacent to $u$ into groups of size $q_i$. Let $E_j$ be the $j$-th group for $t+1 \leq j \leq d$.

Let $p_1, \ldots, p_d$ be the port numbers of the $d$ neighbors of $u$ in $\hat{T}_0$. Note that $p_1, \ldots, p_d$ form a permutation of numbers in $\{1, \ldots, d\}$. Define $B_k$ as the set of integers in $[(k-1)q_i + 1, kq_i]$. For $1 \leq j \leq d$, assign to $E_j$ port numbers from $B_{p_j}$ arbitrarily. Assign all the other port numbers arbitrarily. This concludes the description of the port numbers in $\hat{M}_T$. The following observation provides the key property of the port assignments in $\hat{M}_T$.

**Observation 6.1.** *Let $w_i$ be a child of $u$ and let $p_i \in \{1, \ldots, d\}$ be the port number in $\hat{T}_0$ of the edge $(u, w_i)$, as seen from $u$. Every port number $p$ of $u$ in $\hat{M}_T$ leading to a subtree rooted at $w_i$ satisfies $\lceil p/q_i \rceil = p_i$.*

**Routing without header rewriting.** Next, we show that header rewriting cannot reduce overall memory per vertex in ancestor-descendant routing. Consider an ancestor-descendant routing scheme $\mathcal{R}'$ which routes on top of an instance $\hat{M}_T$ in $\mathcal{T}'$. Let $u$ be a source vertex and $v$ a destination. Initially, the header contains only label$(v)$. Let $w$ be the first vertex on the routing path from $u$ to $v$. Since $\mathcal{R}'$ is a valid ancestor-descendant routing scheme and $w$ is an ancestor of $v$, it is possible to route from $w$ to $v$ with $w$ as a source and $v$ as a destination. In this case, the routing algorithm commences at $w$ and the header contains only label$(v)$. Since the routing scheme has stretch 1, vertex $w$ will never be visited again. In other words, rewriting the header at vertex $u$ does not help in ancestor-descendant routing.

**Reduction to routing in trees.** Let $\hat{T}_0$ be an instance in $\mathcal{T}$ and let $\hat{M}_T$ be the corresponding instance in $\mathcal{T}'$. Our goal is to define a transformation of an ancestor-descendant routing scheme $\mathcal{R}'$ for $\hat{M}_T$ into an ancestor-descendant routing scheme $\mathcal{R}$ for $\hat{T}_0$, which uses additional $O(\log n') = O(\log n)$ bits per vertex when restricting to query pairs that exist in $\hat{T}_0$. Consider an internal vertex $u$ at height $i$ in $\hat{M}_T$ and its descendant (non-dummy vertex) $v$. Let table$'(u)$ be the routing table of $u$ and label$'(v)$ be the label of $v$ in $\mathcal{R}'$. Define label$(v) :=$ label$'(v)$ and let table$(u)$ be a concatenation of table$'(u)$ and a binary encoding of $q_i$. The number of bits required to store $q_i$ is $O(\log n') = O(\log n)$. Let $\mathcal{R}(\text{table}(u), \text{label}(v)) := \lceil \frac{\mathcal{R}'(\text{table}'(u), \text{label}'(v))}{q_i} \rceil$. This concludes the description of $\mathcal{R}$.

We argue that $\mathcal{R}$ is a valid routing scheme for $T$. It suffices to prove that $\mathcal{R}$ outputs the correct port number. Let $p$ be the port number leading to the next vertex on the routing path from $u$ down to $v$. We want to prove that $\mathcal{R}(\text{table}(u), \text{label}(v)) = p$. From Observation 6.1, we know that $\lceil \frac{\mathcal{R}'(\text{table}'(u), \text{label}'(v))}{q_i} \rceil = p'$, where the $p'$ is the port number in $\hat{T}_0$ leading to the child of $u$ which is the root of the subtree where $v$ belongs. Hence, the routing algorithm in $\hat{T}_0$ proceeds at the correct child.

In conclusion, we described a way to convert a routing scheme for $\hat{M}_T$ into a routing scheme in $\hat{T}_0$ which uses $\Omega(\log n)$ additional bits. In [FG02] it is proved that $\mathcal{T}$ contains an instance in which some vertex requires $\Omega(\log^2 n / \log\log n)$ bits of memory. This means that there is an instance in $\mathcal{T}'$ which requires $\Omega(\log^2 n/\log\log n) = \Omega(\log^2 n'/\log\log n')$ bits of memory.

## References


[AAK+06] Serge Abiteboul, Stephen Alstrup, Haim Kaplan, Tova Milo, and Theis Rauhe. Compact labeling scheme for ancestor queries. *SIAM J. Comput.*, 35(6):1295–1309, 2006. 19

[ADM+95] S. Arya, G. Das, D. M. Mount, J. S. Salowe, and M. Smid. Euclidean spanners: Short, thin, and lanky. In *Proceedings of the Twenty-seventh Annual ACM Symposium on Theory of Computing*, STOC '95, pages 489–498, 1995. 1, 2, 5

[AS87] Noga Alon and Baruch Schieber. Optimal preprocessing for answering on-line product queries. Technical Report TR 71/87, Tel Aviv University, 1987. 1, 2, 4

[AS24] Noga Alon and Baruch Schieber. Optimal preprocessing for answering on-line product queries. *CoRR*, abs/2406.06321, 2024. 7, 9, 14

[BFN19] Yair Bartal, Arnold Filtser, and Ofer 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. 5

[BLMN04] Yair Bartal, Nathan Linial, Manor Mendel, and Assaf Naor. Some low distortion metric ramsey problems. *Discrete & Computational Geometry*, 33(1):27–41, July 2004. 16

[BRS25] Kevin Buchin, Carolin Rehs, and Torben Scheele. Geometric spanners of bounded tree-width. In *SoCG*, volume 332 of *LIPIcs*, pages 26:1–26:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2025. 6

[BTS94] Hans L. Bodlaender, Gerard Tel, and Nicola Santoro. Trade-offs in non-reversing diameter. *Nord. J. Comput.*, 1(1):111–134, 1994. 1, 2, 14

[CCC+25] Hsien-Chih Chang, Vincent Cohen-Addad, Jonathan Conroy, Hung Le, Marcin Pilipczuk, and Michal Pilipczuk. Embedding planar graphs into graphs of treewidth $O(\log^3 n)$. In *SODA*, pages 88–123. SIAM, 2025. 2

[CCL+23] H. Chang, J. Conroy, H. Le, L. Milenković, S. Solomon, and C. Than. Covering planar metrics (and beyond): $O(1)$ trees suffice. In *The 64th Annual Symposium on Foundations of Computer Science*, FOCS '23, pages 2231–2261, 2023. 5

[CCL+24] H. Chang, J. Conroy, H. Le, L. Milenković, S. Solomon, and C. Than. Shortcut partitions in minor-free graphs: Steiner point removal, distance oracles, tree covers, and more. In *The 2024 Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '24, page 5300–5331, 2024. 5

[CCL+25] Hsien-Chih Chang, Jonathan Conroy, Hung Le, Shay Solomon, and Cuong Than. Light tree covers, routing, and path-reporting oracles via spanning tree covers in doubling graphs. In *STOC*, pages 2257–2268. ACM, 2025. 5, 20

[CDKX25] Derek G. Corneil, Feodor F. Dragan, Ekkehard Köhler, and Yang Xiang. Lower bounds on collective additive spanners. *CoRR*, abs/2504.18508, 2025. 6

[CFKL20] Vincent Cohen-Addad, Arnold Filtser, Philip N. Klein, and Hung Le. On light spanners, low-treewidth embeddings and efficient traversing in minor-free graphs. In *FOCS*, pages 589–600. IEEE, 2020. 2

[CG06] H. T.-H. Chan and A. Gupta. Small hop-diameter sparse spanners for doubling metrics. In *Proc. of 17th SODA*, pages 70–78, 2006. 16

[Cha87] Bernard Chazelle. Computing on a free tree via complexity-preserving mappings. *Algorithmica*, 2(1):337–361, 1987. 1, 2

[CLPP23] Vincent Cohen-Addad, Hung Le, Marcin Pilipczuk, and Michal Pilipczuk. Planar and minor-free metrics embed into metrics of polylogarithmic treewidth with expected multiplicative distortion arbitrarily close to 1. In *FOCS*, pages 2262–2277. IEEE, 2023. 2

[DFG08] Feodor F. Dragan, Fedor V. Fomin, and Petr A. Golovach. Spanners in sparse graphs. In *ICALP (1)*, volume 5125 of *Lecture Notes in Computer Science*, pages 597–608. Springer, 2008. 6

[FG02] Pierre Fraigniaud and Cyril Gavoille. A space lower bound for routing in trees. In *STACS*, volume 2285 of *Lecture Notes in Computer Science*, pages 65–75. Springer, 2002. 20, 21, 22

[FGvL11] Fedor V. Fomin, Petr A. Golovach, and Erik Jan van Leeuwen. Spanners of bounded degree graphs. *Inf. Process. Lett.*, 111(3):142–144, 2011. 6

[FL22a] Arnold Filtser and Hung Le. Locality-sensitive orderings and applications to reliable spanners. In *STOC*, pages 1066–1079. ACM, 2022. 16

[FL22b] Arnold Filtser and Hung Le. Low treewidth embeddings of planar and minor-free metrics. In *FOCS*, pages 1081–1092. IEEE, 2022. 1, 2, 3, 4, 9

[KLMS22] Omri 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 Philipp Woelfel, editors, *Proc. PODC*, pages 151–162. ACM, 2022. 1, 2, 3, 5

[Le23] H. Le. Shortcutting trees, 2023. `https://minorfree.github.io/tree-shortcutting/`. 1, 3

[MN07] Manor Mendel and Assaf Naor. Ramsey partitions and proximity data structures. *Journal of the European Mathematical Society*, 9(2):253–275, 2007. 5

[NS07] G. Narasimhan and M. Smid. *Geometric Spanner Networks*. Cambridge University Press, 2007. 7, 9

[NW64] Crispin St. John Alvah Nash-Williams. Decomposition of finite graphs into forests. *Journal of the London Mathematical Society*, 1(1):12–12, 1964. 6

[SE14] Shay Solomon and Michael Elkin. Balancing degree, diameter, and weight in euclidean spanners. *SIAM J. Discret. Math.*, 28(3):1173–1198, 2014. 3

[Sol13] Shay Solomon. Sparse euclidean spanners with tiny diameter. *ACM Trans. Algorithms*, 9(3):28:1–28:33, 2013. 9, 14

[Yao82] A. C. Yao. On constructing minimum spanning trees in $k$-dimensional spaces and related problems. *SIAM Journal on Computing*, 11(4):721–736, 1982. 1, 2