Random-Shift Revisited: Tight Approximations for Tree Embeddings and ℓ_1-Oblivious Routings

Rasmus Kyng [*†]
ETH Zurich
kyng@inf.ethz.ch

Maximilian Probst Gutenberg[*]
ETH Zurich
maximilian.probst@inf.ethz.ch

Tim Rieder
ETH Zurich
tim.rieder@inf.ethz.ch

Abstract

We present a new and surprisingly simple analysis of random-shift decompositions—originally proposed by Miller, Peng, and Xu [SPAA'13]: We show that decompositions for exponentially growing scales $D = 2^0, 2^1, \ldots, 2^{\log_2(\mathrm{diam}(G))}$, have a tight *constant* trade-off between distance-to-center and separation probability *on average* across the distance scales – opposed to a necessary $\Omega(\log n)$ trade-off for a single scale.

This almost immediately yields a way to compute a tree T for graph G that preserves all graph distances with expected $O(\log n)$-stretch. This gives an alternative proof that obtains tight approximation bounds of the seminal result by Fakcharoenphol, Rao, and Talwar [STOC'03] matching the $\Omega(\log n)$ lower bound by Bartal [FOCS'96]. Our insights can also be used to refine the analysis of a simple ℓ_1-oblivious routing proposed in [FOCS'22], yielding a tight $O(\log n)$ competitive ratio.

Our algorithms for constructing tree embeddings and ℓ_1-oblivious routings can be implemented in the sequential, parallel, and distributed settings with optimal work, depth, and rounds, up to polylogarithmic factors. Previously, fast algorithms with tight guarantees were not known for tree embeddings in parallel and distributed settings, and for ℓ_1-oblivious routings, not even a fast sequential algorithm was known.

[*]The research leading to these results has received funding from grant no. 200021 204787 of the Swiss National Science Foundation.

[†]The research leading to these results has received funding from the starting grant "A New Paradigm for Flow and Cut Algorithms" (no. TMSGI2 218022) of the Swiss National Science Foundation.

1 Introduction

Distances in graphs lie at the heart of numerous applications in computer science and related fields. Whether modeling communication in social networks [DGPW14, BKE+24], finding routings of almost optimal congestion in networks [Räc08], or analyzing biological networks [PSM+11], efficient distance queries are indispensable. More recently, there has been a rapid increase in interest in optimal transport for machine learning purposes (domain adaptation and generative models, text embeddings) [LR18, ACB17], computer graphics [LCCS18], and analyzing biological data [SST+19], which can be solved in linear time on trees [CTW24].

In many practical scenarios, approximate distances are often sufficient, and rapid query responses are prioritized over exact computations. A particularly elegant solution to increasing speed and drastically reducing storage is the use of *probabilistic tree embeddings*. Given input graph $G = (V, E, w : E \mapsto [1, W])$ with $n := |V|$ and $m := |E|$ and W polynomially-bounded in n. A tree T drawn from some distribution $\mathcal{T}$ is an α-approximate probabilistic tree embedding of G if T preserves all distances in expectation by at most an α-factor, i.e. for any $u, v \in V$, $\mathbb{E}_{T\sim\mathcal{T}}[\text{dist}_T(u, v)] \leq \alpha \cdot \text{dist}_G(u, v)$[1] and $\text{dist}_T(u, v) \geq \text{dist}_G(u, v)$. Probabilistic tree embeddings yield a particularly simple representation of the distances of the underlying graph or metric. Further, many (NP-)hard problems on graphs connected to cuts, network flow, metric labeling, buy-at-bulk network design, and vertex cover become tractable on trees [CKNZ01, KT02, AA97], thus yielding $O(\alpha)$-approximate algorithms to many important graph problems.

Probabilistic tree embeddings. The notion of probabilistic tree embeddings was first studied in [AA97]. The problem of finding good tree embeddings has since received considerable attention. In [AA97], a first algorithm was given achieving an $O(e^{\sqrt{\log n \log\log n}})$ approximation. Since, a long line of work (see [RR98, AKPW95, Bar96, Bar98, FRT03, Bar04] and the references therein) first yielded an algorithm that achieves polylogarithmic approximation [Bar96] and finally culminated in the seminal result by Fakcharoenphol, Rao, and Talwar [FRT03] giving a tree embedding with tight approximation $\Theta(\log n)$. In [Bar04], an $O(\log n)$-approximate tree embedding is given via a proof that condenses the structural insights from [FRT03]. Tree embeddings obtained via their algorithm or some variation have since been referred to as FRT trees. To date, FRT trees are widely popular as the algorithm is reasonably simple and the analysis is insightful. They have also been taught in various undergraduate courses on advanced algorithms.

ℓ_1-oblivious routings. An ℓ_1-oblivious routing is a matrix $\boldsymbol{A} \in \mathbb{R}^{E\times V}$ that given any demand $\boldsymbol{d} \perp \mathbf{1}$, i.e. $\boldsymbol{d}^\top \mathbf{1} = 0$, over the vertices yields a flow $\mathbf{f} = \boldsymbol{A}\mathbf{d}$ that routes the demand. The competitive ratio of an oblivious routing $\boldsymbol{A}$ is the worst ratio achieved over all demands $\mathbf{d}$ between the ℓ_1-cost[2] of the flow $\mathbf{f}$ and the optimal ℓ_1-cost $\text{OPT}(\mathbf{d})$ of any flow routing demand $\mathbf{d}$. Today, ℓ_1-oblivious routings are also an essential building block in distance-based algorithms that employ convex optimization, as the occurring subproblem can be solved by few left- and right-multiplies with $\boldsymbol{A}$ (see [She17, Li20, ZGY+22, REGH22, Zuz23, Fox24]).

Note that for any tree T, we can construct a matrix $\boldsymbol{A}_T$ such that $\mathbf{d}$ is routed optimally on T. Thus, for fixed $\boldsymbol{d}$, an α-approximate probabilistic tree embedding T yields that $\boldsymbol{A}_T\boldsymbol{d}$ has expected

[1] We note that a deterministic notion of probabilistic tree embeddings has been considered where the goal is to find a tree that minimizes the average stretch among all edges

[2] This is also often referred to as the cost of a *transshipment flow*.

ℓ_1-cost α. It follows that for FRT trees T being sampled from distribution $\mathcal{T}$ with probability p_T, we have that $\boldsymbol{A} = \sum_{T \in \mathcal{T}} p_T \cdot \boldsymbol{A}_T$ is $O(\log n)$ competitive for every demand, and thus, an ℓ_1-oblivious routing of quality $O(\log n)$. This is tight by the lower bounds on probabilistic tree embeddings. Alternatively, a construction by Englert and Räcke [ER09] achieves a tight competitive ratio.

Low-diameter decompositions and probabilistic tree embeddings. Given a graph G and a diameter D, a low-diameter decomposition is a partition $\mathcal{X}$ of the vertex set into clusters $X \in \mathcal{X}$ that separate any two vertices $u, v \in V$ with probability at most $O(\text{dist}_G(u,v)/D)$ while every cluster has weak diameter $diam_G(X) = \max_{x,y \in X} \text{dist}_G(x,y)$ at most $O(D \log n)$. The trade-off between separation probability and diameter is known to be tight [Bar96].

Probabilistic tree embeddings are intimately connected to low-diameter decompositions: a hierarchy of low-diameter decompositions $\mathcal{A}_0 = \{\{v\} \mid v \in V\}, \mathcal{A}_1, \ldots, \mathcal{A}_L = \{V\}$ for $L = \lceil \log_2(nW) \rceil$ where each $\mathcal{A}_l$ is a low-diameter decomposition of G for parameter 2^l can be used almost directly to induce a probabilistic tree. That is, letting $\mathcal{A}_{\geq l}$ be the coarsest common refinement of partitions $\mathcal{A}_l, \mathcal{A}_{l+1}, \ldots, \mathcal{A}_L$, we can construct a tree T that has a node associated with each cluster C in a partition $\mathcal{A}_{\geq l}$ and for $l < L$ and edge to the node associated with cluster $C \subseteq C' \in \mathcal{A}_{\geq l+1}$ of weight $diam_G(C')$. The approximation factor achieved is $O(\log^2 n)$ with one log-factor stemming from the (tight) trade-off between separation probability and the diameter bound on each level, and the second log-factor from the number of levels.

In [FRT03], a hierarchical low-diameter decomposition is developed such that for every vertex $u \in V$, the trade-off between separation probabilities of u and diameter of clusters containing u costs a log-factor across **all** levels. Formally, letting $\mathcal{A}_l(u)$ denote the cluster of u in $\mathcal{A}_l$, this property can be succinctly summarized as $\sum_{0 \leq l \leq L} \frac{diam_G(\mathcal{A}_l(u))}{2^l} = O(\log n)$ (which implies $\sum_{0 \leq l \leq L} \frac{diam_G(\mathcal{A}_{\geq l}(u))}{2^l} = O(\log n)$).

Random-shift decompositions. In this article, we consider a different type of low-diameter decompositions than [FRT03]: random-shift decompositions – first suggested by Miller, Peng and Xu [MPX13]. Random-shift Decompositions are popular due to their simplicity and amenability to various computational settings. These decompositions are computed by the algorithm given in Algorithm 1. The algorithm first samples a negative delay for each vertex $u \in V$ independently from the exponential distribution with mean D. It then computes a clustering around vertices with large negative delay by running a single-source shortest path (SSSP) computation.

Algorithm 1: RandomShiftDecomposition($G = (V, E, w : E \mapsto [1, W], D)$)

for *each vertex* $u \in V$ **do**
  Pick δ^u independently from the exponential distribution with mean D.
Let G^s be the graph obtained from adding dummy source s to G and an edge (s, v) of weight $\max_{w \in V} \delta^w - \delta^v$.
Call an exact SSSP procedure on G^s from s and let T^s be the shortest path tree obtained rooted at s.
for *each vertex* $u \in V$ *that is a child of* s *in* T^s **do**
  C^u is the set of all vertices in the subtree of T^s rooted at u.
  foreach $v \in C^u$ **do** Center(v) $\leftarrow u$.;
return ($\mathcal{C} = \bigcup_{u \in V} \{C^u\} \setminus \{\emptyset\}$, Center)

A simple but clever analysis (see [MPX13]) yields that the separation probability for any u, v can be bounded by $O(\text{dist}_G(u,v)/D)$ as desired, and it is easy to glean from the exponential distribution that no negative delay exceeds $O(D \log n)$ w.h.p. which yields the desired diameter bound of $O(D \log n)$.

Let $\mathcal{C}_0 = \{\{v\} \mid v \in V\}$ and each singleton cluster has the vertex as center, the sets $\mathcal{C}_l$ for $0 < l < L$ being obtained by calling Algorithm 1 with parameter 2^l and $\mathcal{C}_L = \{V\}$ where the center of V is an arbitrarily chosen vertex in V. Then, we call $\mathcal{C}_0, \mathcal{C}_1, \ldots, \mathcal{C}_L$ a *hierarchical random-shift decomposition*. Naturally, we can derive a tree embedding T as described above from these decompositions with approximation $O(\log^2 n)$. Later, [BEL20] showed how to compute such hierarchical random shift decompositions in parallel using approximate SSSP computations.

Because hierarchical random-shift decompositions are simple, efficient, and parallelization-friendly, it is natural to ask if they can yield even stronger results. However, a priori, this might seem unlikely, as it is easy to construct examples where for fixed $u \in V$, the diameter of every cluster $\mathcal{C}_l(u)$ for $0 < l < L$ is $\Omega(2^l \log n)$ and thus one can seemingly not hope for a tight tree embedding induced by these decompositions.

A slightly tighter tree embedding construction. On further inspection, one can slightly tighten the tree construction used above by choosing centers: Let $\textsc{Center}_0, \textsc{Center}_1, \ldots, \textsc{Center}_L$[3] be the associated center functions for the hierarchical random-shift decompositions $\mathcal{C}_0, \mathcal{C}_1, \ldots, \mathcal{C}_L$ as above where $\textsc{Center}_0$ maps each vertex to itself, and $\textsc{Center}_L$ maps all vertices to an arbitrary fixed vertex $r \in V$.

When forming the tree induced by the (refinements of) $\mathcal{C}_0, \mathcal{C}_1, \ldots, \mathcal{C}_L$, one can now set the weight of edges (C, C') where $C \in \mathcal{C}_{\geq l}, C \subseteq C' \in \mathcal{C}_{\geq l+1}$ equal to $\text{dist}_G(\textsc{Center}_l(v), \textsc{Center}_{l+1}(v))$ for some $v \in C$ instead of using the rather loose upper bound of $diam_G(C')$. Further, by consistency of the center function and the triangle inequality, we then have for every $w \in C$, that

$$\begin{aligned}\text{dist}_G(\textsc{Center}_l(v), \textsc{Center}_{l+1}(v)) &= \text{dist}_G(\textsc{Center}_l(w), \textsc{Center}_{l+1}(w)) \\ &\leq \text{dist}_G(\textsc{Center}_l(w), w) + \text{dist}_G(w, \textsc{Center}_{l+1}(v)).\end{aligned}$$

This yields an $O(\log n)$-approximate tree embedding when the trade-off between separation probabilities and distances to cluster centers is constant on average, and thus $O(\log n)$ over all distance scales.

1.1 Our Contribution

In this article, we show that for hierarchical random-shift decompositions, the trade-off between separation probabilities and distances to centers is indeed bounded by $O(\log n)$.[4] We summarize this result in the theorem below.

Theorem 1.1. *Given graph $G = (V, E, w : E \mapsto [1, W])$, let $\mathcal{C}_0, \mathcal{C}_1, \ldots, \mathcal{C}_L$ be a hierarchical random-shift decomposition as described above. Then, for every $0 \leq l \leq L$ and $v \in V$, let $\mathcal{C}_l(v)$ be the cluster containing v in $\mathcal{C}_l$. Then, we have*

[3] Throughout, all center functions are consistent, meaning all vertices in the same cluster have the same center.

[4] It was since pointed out that a similar result was derived in [CD21], Theorem 2. This result, in turn, tightens the analysis from [HW16] by an $O(\log \log n)$ factor.

1. *for any* $v \in V$,
$$\mathbb{E}\left[\sum_{0\le l\le L} \frac{\mathrm{dist}_G(v, \textsc{Center}_l(v))}{2^l}\right] \le O(\log n).$$

2. *for any* $0 \le l \le L$, $u, v \in V$, $\mathbb{P}[\mathcal{C}_l(u) \ne \mathcal{C}_l(v)] = O\left(\frac{\mathrm{dist}_G(u,v)}{2^l}\right)$.

Remark 1.2. *For $0 \le l \le L$, we define $\mathcal{C}_{\ge l}$ to be the coarsest common refinement of $\mathcal{C}_l, \mathcal{C}_{l+1}, \ldots, \mathcal{C}_L$. For $v \in C \in \mathcal{C}_{\ge l}$, we define* $\textsc{Center}_{\ge l}(v) = \arg\min_{r\in C} \mathrm{dist}(r, \textsc{Center}_l(v))$.

Then, the hierarchy $\mathcal{C}_0, \mathcal{C}_1, \ldots, \mathcal{C}_L$ along with functions $\textsc{Center}_{\ge 0}, \textsc{Center}_{\ge 1}, \ldots, \textsc{Center}_{\ge L}$ satisfies the properties above and is refining. Therefore, we say it is a refining hierarchical random-shift decomposition.

Since the sets in $\mathcal{C}_{\ge 0}, \mathcal{C}_{\ge 1}, \ldots, \mathcal{C}_{\ge L}$ form a laminar family, they naturally induce a tree T, and it is not hard to show that T forms an $O(\log n)$-approximate probabilistic tree embedding.

We then show that a hierarchical *approximate* random-shift decomposition $\widehat{\mathcal{C}}_0, \widehat{\mathcal{C}}_1, \ldots, \widehat{\mathcal{C}}_L$ as suggested in [BEL20] yields almost the same properties as stated in Remark 1.2: the only difference being that each clustering only forms a subpartition, and each vertex v is only clustered with probability at least $1/2$ (and thus the first property only applies to levels where v was clustered and thus a center for v exists). We show further that, up to constant factors, the refinements $\widehat{\mathcal{C}}_{\ge 0}, \widehat{\mathcal{C}}_{\ge 1}, \ldots, \widehat{\mathcal{C}}_{\ge L}$ satisfy the same properties as its exact counterpart, given in Remark 1.2.

Since the algorithm from [BEL20] only requires approximate SSSP as a primitive, it can be implemented optimally up to logarithmic factors in the sequential, parallel, distributed, and semi-streaming settings[5]. This implies fast algorithms for probabilistic tree embeddings in these settings.

Theorem 1.3 (Main Result for Probabilistic Tree Embeddings)**.** *Given graph $G = (V, E, w)$, there is a randomized algorithm that returns a $O(\log n)$-approximate tree embedding T of G.*

The algorithm can be implemented to run

1. *in time $O(m \log n)$ in the sequential model.*
2. *in polylogarithmic depth and almost-linear work in the PRAM model.*
3. *in $\tilde{O}(\sqrt{n} + \textsc{HopDiam}(G))$[6] rounds with message complexity $\tilde{O}(m)$ in the CONGEST model where $\textsc{HopDiam}(G)$ is the unweighted (i.e. hop) diameter of the graph G.*
4. *in $\tilde{O}(1)$ rounds in the semi-streaming model with high probablility.*

The theorem above recovers the fastest runtime of any sequential algorithm to construct FRT trees [BGS17] while being significantly simpler; the first parallel algorithm that simultaneously achieves work $\tilde{O}(m)$ and $\tilde{O}(1)$ depth; and the first distributed algorithm matching the lower bound from [DSHK+11] on the number of rounds up to polylogarithmic factors. In the latter two settings, this yields a polynomial improvement for non-dense graphs over the state of the art [BGSS20, FL18, GL14].

[5] We refer the reader to [BEL20] for precise definitions of the PRAM, CONGEST and semi-streaming models. We further point the reader to [REGH22] for information about a simpler minor-aggregation model that can be implemented efficiently in PRAM and CONGEST models.

[6] In this article, we let n denote the number of vertices of the input graph and write $\tilde{O}(f(k))$ to mean $O(f(k)\log^c(n))$ for any constant c.

Further, we show that the ℓ_1-oblivious routings from [REGH22] have $O(\log n)$-approximation. While we essentially use their construction, we give a different proof that leverages the insights from Theorem 1.1 to obtain the tight approximation factor.

Theorem 1.4 (Main Result for ℓ_1-Oblivious Routing)**.** *Given graph $G = (V, E, w)$, there is a randomized algorithm that returns an $O(\log n)$-approximate ℓ_1-oblivious routing $\boldsymbol{A}$ of G w.h.p.*

The algorithm can be implemented to run

1. *in time $\tilde{O}(m)$ in the sequential model.*
2. *in polylogarithmic depth and almost-linear work in the PRAM model.*
3. *in $\tilde{O}(\sqrt{n} + \textsc{HopDiam}(G))$ rounds with message complexity $\tilde{O}(m)$ in the CONGEST model.*

Notably, this yields the first $\tilde{O}(m)$ algorithm to construct ℓ_1-oblivious routings of asymptotically optimal competitive ratio! While $\tilde{O}(m)$ time algorithms were previously known, all of them obtained rather large subpolynomial or polylogarithmic factors in the competitive ratio.

Roadmap. We defer a detailed discussion of related work to Section A. In Section 3, we prove Theorem 1.1. In Section 4, we prove that almost the same guarantees hold for approximate random-shift decompositions as suggested in [BEL20]. Finally, in Section 5, we analyze the resulting tree embeddings, and in Section 6, we give a novel analysis of the ℓ_1-oblivious routing construction suggested in [REGH22] yielding an ℓ_1-oblivious routing with tight competitive ratio.

2 Preliminaries

Graphs. We denote graphs as tuples $G = (V, E, w : E \mapsto [1, W])$ with $n := |V|$ and $m := |E|$. In this article, all graphs are undirected and weighted. For a subset $A \subset V$, we use $G[A]$ to denote the vertex-induced graph and $E[A]$ to denote the set of edges with both endpoints in A, i.e. $E[A] = \{\{u, v\} \in E, u, v \in A\}$. For disjoint subsets $A, B \subset V$, we denote by $E(A, B)$ edges in E with exactly one endpoint in A and B. We denote by $dist_G(s, t)$ the distance in G under w from s to t. We let $B_G(v, r)$ denote the closed ball centered at v with radius r. We often omit the subscript G when the underlying graph G is clear.

Exponential Distribution. We recall that the exponential distribution with mean D, denoted by $\textsc{Exp}(D)$, has *probability density function* (pdf) $p(x) = e^{-x/D}/D$ and *cumulative density function* (cdf) $F(x) = 1 - e^{-x/D}$.

The exponential distribution has the memoryless property, i.e. for $X \sim \textsc{Exp}(D)$, we have $\mathbb{P}[X > x + y | X > y] = \mathbb{P}[X > x]$ for all $x, y \geq 0$. Note further, that the exponential distribution satisfies that $X \sim \textsc{Exp}(1)$ iff $D \cdot X \sim \textsc{Exp}(D)$ for any $D > 0$.

Chernoff Bounds. We use the classic Chernoff bound for i.i.d. random variables $X_1, X_2, \ldots, X_n \in [0, 1]$, that for $\mu = \sum_i X_i$, we have for every $\delta > 0$, that $\mathbb{P}[|X - \mu| > \delta\mu] < 2 \cdot e^{-\delta^2\mu/3}$.

We also use the Chernoff bound for i.i.d. random variables $X_1, X_2, \ldots, X_n \sim \textsc{Exp}(1)$ which yields that for $X = \sum_i X_i$, we have for any $\delta > 0$, that $\mathbb{P}[X > (2 + \delta)n] < e^{-\delta n/2}$. Note that this bound does not make any assumptions on the values that variables X_i can take. The bound can be derived straightforwardly via the moment-generating function of the exponential distribution. For a lack of an accessible resource, we give a short proof in Section B.

Logarithms. We denote by $\log(x)$ the natural logarithm of x and by $\lg(x)$ the logarithm of x with base 2.

3 Random-Shift Decompositions via Exact SSSP

In this section, we give a proof of Theorem 1.1 for the case when $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_{L-1}$ are computed via the standard random-shift algorithm from Algorithm 1.

Correctness of the first property via a simple ball growing process. At the heart of our new analysis lies a simple ball growing process over levels l. Fixing the vertex $v \in V$, we define the process by a sequence of integers r_l such that balls $B(v, r_l 2^l)$ are growing monotonically.

Let $r_0 := 1$ with $|B(v, r_0 2^0)| \geq 1$ and let $r_l := r_{l-1}/2 + \Delta_l + 2$, where we choose Δ_l ($l \geq 1$) as the largest integer such that growing the radius of the ball can be paid for by an exponential growth in the number of vertices in it, namely

$$\Delta_l := \max \xi \text{ satisfying } |B(v, (r_{l-1}/2 + \xi + 2)2^l)| \geq e^{\xi/2} \cdot |B(v, r_{l-1}2^{l-1})|. \tag{1}$$

Note that $r_l 2^l = (r_{l-1}/2 + \Delta_l + 2)2 \cdot 2^{l-1} > r_{l-1}2^{l-1}$, thus $r_l 2^l$ grows monotonically as claimed.

In the remainder of this section, we prove the following two lemmas. The first lemma says that the distance to the center is well captured by $r_l 2^l$ on level l (in expectation). The second yields that the sum of r_l's is small, and thus, since r_l can be thought of as the loss-per-level, we have a small loss overall. Combined, these lemmas yield Property 1 as a corollary.

Lemma 3.1 (Distances-to-Center)**.** *For any* $0 \leq l \leq L$, $\mathbb{E}[\text{dist}(v, \textsc{Center}_l(v))] = O(r_l \cdot 2^l)$.

Lemma 3.2 (Sum-of-Deviations)**.** $\sum_{l=0}^{L} r_l \in O(\log n)$.

Bounding the distances to centers (Proof of Lemma 3.1). Let us fix any level $0 < l < L$ (for $l \in \{0, L\}$ the proof is trivial). We let the variables δ^u and C^u refer to the values that these variables take in Algorithm 1 when computing $\mathcal{C}_l$.

For every $i \geq 0$, we let $B_i = B(v, (r_l + i)2^l)$ and define $I_i = B_{i+2} \setminus B_{i+1}$. We can now rewrite the expectation as

$$\mathbb{E}[\text{dist}(v, \textsc{Center}_l(v))] = \sum_{w \in V} \mathbb{P}[v \in C^w] \cdot \text{dist}(w, v) \leq O(r_l \cdot 2^l) + \sum_{i \geq 1} \sum_{w \in I_i} \mathbb{P}[v \in C^w] \cdot (i+1) \cdot 2^l.$$

It remains to bound the probabilities that a vertex in I_i becomes the center of v. Note that a vertex $w \in I_i$ cannot become a center if for some $u \in B_0$, $\delta^u + i \cdot 2^l > \delta^w$, since then v prefers u over w as

a center. We obtain for every $w \in I_i$

$$\begin{aligned}
\mathbb{P}[v \in C^w] &\leq \mathbb{P}[\max_{u \in B_0} \delta^u < \delta^w - i \cdot 2^l] \\
&= \int_{x=i\cdot 2^l}^{\infty} \frac{1}{2^l} e^{-x/2^l} \cdot \mathbb{P}[\max_{u \in B_0} \delta^u + i \cdot 2^l < x] dx \\
&= \int_{x=i\cdot 2^l}^{\infty} \frac{1}{2^l} e^{-x/2^l} \cdot (1 - e^{-x/2^l + i})^{|B_0|} dx \\
&= \int_{x=0}^{\infty} \frac{1}{2^l} e^{-x/2^l - i} \cdot (1 - e^{-x/2^l})^{|B_0|} dx \\
&= e^{-i} \int_{x=0}^{\infty} \frac{1}{2^l} e^{-x/2^l} \cdot (1 - e^{-x/2^l})^{|B_0|} dx
\end{aligned} \tag{2}$$

$$\begin{aligned}
&\leq e^{-i} \int_{x=0}^{\infty} \frac{1}{2^l} e^{-x/2^l} \cdot e^{-|B_0| \cdot e^{-x/2^l}} dx \\
&= e^{-i} \cdot \left[\frac{e^{-|B_0| \cdot e^{-x/2^l}}}{|B_0|} \right]_{x=0}^{\infty} \\
&= e^{-i} \cdot \left(\frac{1}{|B_0|} - \frac{e^{-|B_0|}}{|B_0|} \right) \\
&< \frac{e^{-i}}{|B_0|}.
\end{aligned}$$

where in the first equality, we compute the probability by integrating over all values x that δ^w can take. At each such value x, we take the density function $\frac{1}{2^l} e^{-x/2^l}$ of the exponential distribution times the probability that the maximum value δ^u over all $u \in B_0$ is smaller than x. Note that we start integrating only over $x \geq i \cdot 2^l$, since for smaller values, δ^w cannot exceed $\max_u \delta^u + i \cdot 2^l$ since $\delta^u \geq 0$.

In the second inequality, we use the probability that the maximum value over all variables δ^u is at most x is given by the cumulative density function at value x to the power $|B_0|$.

We substitute the value of x in the third equality to simplify the expression, and pull out the factor e^{-i} from the integral in the forth. We then use $1 + y \leq e^y$ in the second inequality, and finally evaluate the integral.

We can therefore now conclude the proof using the above derivation combined with the bound

$|I_i| < e^{(i+2)/2}|B_0|$ that follows from the growth process and $I_i \subseteq B_{i+2}$, which yields

$$\begin{aligned}
\mathbb{E}[\mathrm{dist}(v, \textsc{Center}_l(v)] &= O(r_l \cdot 2^l) + \sum_{i\geq 1}\sum_{w\in I_i} \mathbb{P}[v \in C^w] \cdot (i+1) \cdot 2^l \\
&\leq O(r_l \cdot 2^l) + \sum_{i\geq 1}\sum_{w\in I_i} \frac{e^{-i}}{|B_0|} \cdot (i+1) \cdot 2^l \\
&= O(r_l \cdot 2^l) + \sum_{i\geq 1} |I_i| \cdot \frac{e^{-i}}{|B_0|} \cdot (i+1) \cdot 2^l \\
&\leq O(r_l \cdot 2^l) + \sum_{i\geq 1} e^{(i+2)/2}|B_0| \cdot \frac{e^{-i}}{|B_0|} \cdot (i+1) \cdot 2^l \\
&\leq O(r_l \cdot 2^l) + e \cdot \sum_{i\geq 1} e^{-i/2} \cdot (i+1) \cdot 2^l \\
&= O(r_l \cdot 2^l)
\end{aligned} \tag{3}$$

by straightforward calculations and using that the last sum is geometric.

We point out that inspection of (3) yields a slightly stronger statement: $\mathrm{dist}(v, \textsc{Center}_l(v))/2^l$ is stochastically dominated by $O(r_l + X_l)$ for some random variable $X_l \sim \textsc{Exp}(1)$.

Bounding the sum of deviations (Proof of Lemma 3.2). From the definition of Δ_l in (1), we have $|B(v, r_L D_L)| \geq e^{\Delta_L/2} \cdot |B(v, r_{L-1} D_{L-1}| \geq e^{\Delta_L/2} \cdot e^{\Delta_{L-1}/2} \cdots |B(v, r_0 D_0)|$. But since there are at most n vertices, we have $n \geq |B(v, r_L D_L)| \geq |B(v, r_0 D_0)| \cdot \prod_{l=1}^{L} e^{\Delta_l/2} \geq e^{\sum_{l=1}^{L} \Delta_l/2}$. Thus, $\sum_{l=1}^{L} \Delta_l \leq 2 \ln n$. We thus get

$$\begin{aligned}
\sum_{l=0}^{L} r_l &= \sum_{l=0}^{L} \left(r_0/2^l + \sum_{i=1}^{l} (\Delta_i + 2)/2^{l-i} \right) && \text{By definition of } r_l \\
&= \sum_{l=0}^{L} r_0/2^l + \sum_{i=1}^{L}\sum_{l=i}^{L} (\Delta_i + 2)/2^{l-i} && \text{Reordering the summation} \\
&\leq \sum_{l=0}^{\infty} r_0/2^l + \sum_{i=1}^{L}\sum_{l'=0}^{\infty} (\Delta_i + 2)/2^{l'} && \text{Using } l' = l - i \\
&= O(\log n).
\end{aligned} \tag{4}$$

Correctness of the second property. The second property, that $\mathbb{P}[\mathcal{C}_l(u) \neq \mathcal{C}_l(v)] = O\left(\frac{\mathrm{dist}(u,v)}{2^l}\right)$ for all $0 \leq l \leq L$ and $u, v \in V$, follows from the definitions of $\mathcal{C}_0$ and $\mathcal{C}_L$ and the following insight first exploited in [MPX13].

Lemma 3.3 (Lemma 4.4 of [MPX13])**.** *Let $d_1 \leq d_2 \leq \cdots \leq d_n$ be arbitrary values and let $\delta_1, \ldots, \delta_n$ be independent random variables picked from* $\mathrm{Exp}(D)$*, and define $X_i = d_i - \delta_i$, and define $X^{(1)} \leq X^{(2)} \leq \ldots \leq X^{(n)}$ to be the sequence of $\{X_i\}$ but sorted by increasing value. Then the probability that $X^{(2)} - X^{(1)} \leq c$ is at most c/D.*

Sketch of proof. We model each value $-X_i = \delta_i - d_i$ as the failure time of a light bulb following an exponential distribution with rate β which is turned on at time $-d_i$. The quantity $X^{(1)} = \min_i X_i$

corresponds to the moment the last bulb burns out, and we want to estimate how close this is to the moment when the second-to-last bulb fails. Because exponential distributions are memoryless, once the second-to-last bulb fails, the remaining bulb's remaining lifetime still follows $\mathrm{Exp}(\beta)$. Therefore, the probability that the last bulb also fails within additional time c is $1 - e^{-c\beta} \leq \beta c$. If the last bulb has not yet been turned on at the moment the second-to-last bulb fails, then the probability of a gap under c can only decrease. Consequently, the overall probability that the time difference between the last two bulb failures is less than c remains bounded by $1 - e^{-c\beta}$. □

For vertex $v \in V$, let d_i be the distance from v to the i-th vertex in the graph G. Let δ^i be the random value associated with this vertex in Algorithm 1. Then, the above result implies via the triangle inequality that $B(v, \mathrm{dist}(u, v))$ is contained in cluster $\mathcal{C}(v)$ with probability at least $\mathrm{dist}(u, v)/(2D)$.

Refining hierarchical random-shift decompositions (Proof of Remark 1.2). We recall that $\mathcal{C}_{\geq l}$ is the coarsest common refinement of $\mathcal{C}_l, \mathcal{C}_{l+1}, \ldots, \mathcal{C}_L$ and for $v \in C \in \mathcal{C}_{\geq l}$, $\textsc{Center}_{\geq l}(v) = \arg\min_{u \in C} \mathrm{dist}(r, \textsc{Center}_l(u))$. For the first property, it suffices to use that $C \subseteq C' \in \mathcal{C}_l$ which implies that all vertices $w \in C$ have the same center $\textsc{Center}_l(v)$, and the triangle inequality to conclude that

$$\begin{aligned} \mathrm{dist}(v, \textsc{Center}_{\geq l}(v)) &\leq \min_{u \in C} \mathrm{dist}(\textsc{Center}_l(v), u) + \mathrm{dist}(\textsc{Center}_l(v), v) \\ &\leq 2 \cdot \mathrm{dist}(\textsc{Center}_l(v), v). \end{aligned} \tag{5}$$

For the second property, note that vertices $u, v \in V$ are separated only in $\mathcal{C}_{\geq l}$ if they are separated in any of the clusters $\mathcal{C}_l, \mathcal{C}_{l+1}, \ldots, \mathcal{C}_L$ and thus with probability at most $\sum_{i=l}^{L} O(\mathrm{dist}(u, v)/2^i) = O(\mathrm{dist}(u, v)/2^l)$.

4 Random-Shift Decompositions via Approximate SSSP

In this section, we show that a tweaked version of Algorithm 1 that only requires calls to approximate SSSP yields the guarantees described in Theorem 1.1. Our algorithm is almost exactly the algorithm from [BEL20], except that we save one level of recursion (this will suffice for us; for the application in [BEL20] this would not yield the desired guarantees). We choose $\epsilon = \frac{1}{40 \log n}$ throughout this section.

Algorithm 2: ApproxRandomShiftDecomposition($G = (V, E, w : E \mapsto [1, W], D)$)

for *each vertex* $u \in V$ **do**
  Pick δ^u independently from the exponential distribution with mean D.
Let G^s be the graph obtained from adding dummy source s to G and an edge (s, v) of weight $\max_{w \in V} \delta^w - \delta^v$.
Call a $(1+\epsilon)$-approximate SSSP procedure on G^s from s and let T^s be the approximate shortest path tree obtained rooted at s.
for *each vertex* $u \in V$ *that is a child of* s *in* T^s **do**
  $\widetilde{C}^u$ is the set of all vertices in the subtree of T^s rooted at u.
  foreach $v \in \widetilde{C}^u$ **do** $\widetilde{\text{Center}}(v) \leftarrow u$.;
  $\widehat{C}^u \leftarrow \widetilde{C}^u \setminus \text{Blur}(G, V \setminus \widetilde{C}^u, D)$.
  foreach $v \in \widehat{C}^u$ **do** $\widehat{\text{Center}}(v) \leftarrow u$.;
return $(\widehat{\mathcal{C}} = \bigcup_{u \in V} \{\widehat{C}^u\} \setminus \{\emptyset\}, \widehat{\text{Center}})$

Algorithm 3: Blur(G, X, D)

$\widehat{X} \leftarrow X$.
for $i = 0, 1, \ldots, \lceil \log_{1/\epsilon}(D) \rceil$ **do**
  Let G^i be the graph obtained from G by contracting $\widehat{X}$ into a super-vertex.
  Let r^i be sampled uniformly from $[0, \epsilon^i D/64]$.
  Call a $(1+\epsilon^2)$-approximate SSSP procedure on G^i from $\widehat{X}$ and let T^i be the approximate shortest path tree obtained rooted at $\widehat{X}$.
  Add to $\widehat{X}$ all new vertices in $B_{T^i}(\widehat{X}, r^i)$
return $\widehat{X}$.

Preliminaries for the analysis. We commonly refer to $\mathcal{C}$ and Center as the objects obtained from running Algorithm 1 for the same parameter and random choices.

We further observe that when all values $\delta^u \leq D \cdot 9 \log n$, then all distances in G_s are at most $D \cdot 10 \log n$ and thus by choice of ϵ at most an additive error of $D/4$ in any computed distance (and also T_s introduces at most additive slack $D/4$ in the distances). Since all of these events hold w.p. $1 - n^{-8}$, we condition on it for the rest of this section.

Each vertex v is clustered with constant probability. We first prove the following statement that trivially implies that every vertex is clustered with at least constant probability.

Lemma 4.1. *For every vertex $v \in V$, Algorithm 2 invoked with parameter D returns $\widehat{\mathcal{C}}$ such that $B(v, D/8)$ is contained in some cluster in $\widehat{\mathcal{C}}$ with probability at least $1/2$.*

Proof. Recall that $\mathcal{C}$ is the clustering obtained from running Algorithm 1 with the same parameter and random choices. Lemma 3.3 tells us that with probability $\geq 1/2$, the path $s \to \text{Center}(v) \rightsquigarrow v$ is $D/2$ shorter than any other path $s \to w \rightsquigarrow v$, and thus $B(v, D/4) \in \mathcal{C}(v)$ as well. But note that since there is at most an additive error $D/16$ in the distances, this event implies that $B(v, D/8 + D/16) \subseteq \widetilde{C}(v)$. Finally, we note that Algorithm 3 only removes vertices $v \in \widetilde{C}(v)$ with

$\text{dist}(v, V \setminus \widetilde{\mathcal{C}}(v)) \leq (1+\epsilon)\sum_{i=0}^{\lceil \log_{1/\epsilon}(D)\rceil} r^i \leq 2\sum_{i=0}^{\infty} \epsilon^i D/64 < D/16$. Thus, with probability at least $1/2$, v is not removed in this process, as desired. □

Correctness of the first property. We follow closely the proof from Section 3: we use the same ball growing process, i.e. the ball w.r.t. the real graph distances. For fixed vertex $v \in V$, this yields the sequence $\{r_l\}_{0 \leq l \leq L}$. Lemma 3.2 remains true since it only used values r_l.

We then show that for every level $0 < l < L$, if v is clustered by $\mathcal{C}_l$, then $\frac{\text{dist}(v, \widehat{\text{CENTER}}_l(v))}{2^l}$ is stochastically dominated by $O(r_l + X_l)$ for some random variable $X_l \sim \text{EXP}(1)$ (that only depends on $\mathcal{C}_l$). Let us now fix any such level $0 < l < L$. For every $i \geq 0$, we let $B_i = B(v, (r_l + i)2^l)$ and define $I_i = B_{i+2} \setminus B_{i+1}$. We can now again rewrite the expectation as

$$\mathbb{E}[\text{dist}(v, \widetilde{\text{CENTER}}_l(v)] = \sum_{w \in V} \mathbb{P}[v \in \widetilde{C}^w] \cdot \text{dist}(w, v) \leq O(r_l \cdot 2^l) + \sum_{i \geq 1} \sum_{w \in I_i} \mathbb{P}[v \in \tilde{C}^w] \cdot (i+1) \cdot 2^l.$$

Finally, we have to slightly adapt our calculation of an upper bound on $\mathbb{P}[v \in \tilde{C}^w]$. Namely, since there is an additive error of up to 2^l in the distance computation, $w \in I_i$ cannot become a center of v if for some $u \in B_0, \delta^u + i \cdot 2^l > \delta^w + 2^l$, i.e. we weaken the inequality in the event by an additive 2^l compared to the proof in Section 3. But this still yields that

$$\begin{aligned}
\mathbb{P}[v \in \tilde{C}^w] &\leq \mathbb{P}[\max_{u \in B_0} \delta^u < \delta^w - (i-1) \cdot 2^l] \\
&= \int_{x=(i-1)\cdot 2^l}^{\infty} \frac{1}{2^l} e^{-x/2^l} \cdot \mathbb{P}[\max_{u \in B_0} \delta^u + (i-1) \cdot 2^l < x] dx \\
&< \frac{e^{-i+1}}{|B_0|}.
\end{aligned} \tag{6}$$

This matches the bound of Equation (2) up to a factor e, allowing us to recover the bound of Equation (3). The rest of the proof is analogous, yielding that $\mathbb{E}[\text{dist}(v, \widetilde{\text{CENTER}}_l(v)]/2^l$ is stochastically dominated by $O(r_l + X_l)$ for some $X_l \sim \text{EXP}(1)$.

Finally, $\mathbb{E}[\text{dist}(v, \widehat{\text{CENTER}}_l(v)]/2^l$ is stochastically dominated by $O(r_l + X_l)$, since $\widehat{C}^u \subseteq \widetilde{C}^u$ and we do not define the distance for vertices with no assigned center.

The first property for refining partitions. Let us denote the event that v is not in a cluster in $\mathcal{C}_j$ by $\mathcal{E}_j$. It remains to observe that $\widehat{\text{CENTER}}_{\geq l}(v)$ is found by projecting the center $\widehat{\text{CENTER}}_{l'}(v)$ for the smallest level $l' \geq l$ where the event $\mathcal{E}_{l'}$ holds. As in (5) and our previous reasoning, we

have by the triangle inequality, that $\operatorname{dist}(v, \widehat{\textsc{Center}}_{\geq l}(v) \leq 2 \cdot \operatorname{dist}(v, \widehat{\textsc{Center}}_{l'}(v))$. We conclude

$$\begin{aligned}
&\mathbb{E}\left[\sum_{0<l<L} \frac{\operatorname{dist}(v, \widehat{\textsc{Center}}_{\geq l}(v))}{2^l}\right] \\
&\leq \sum_{0<l<L} \sum_{0\leq i\leq L-l} \mathbb{P}[\cap_{l\leq j<l+i} \mathcal{E}_j \wedge \neg\mathcal{E}_{l+i}] \cdot 2 \cdot \mathbb{E}[\operatorname{dist}(v, \widehat{\textsc{Center}}_{l+i}(v))/2^{l+i} \mid \neg\mathcal{E}_{l+i}] \\
&\leq \sum_{0<l<L} \sum_{0\leq i\leq L-l} \prod_{l\leq j<l+i} \mathbb{P}[\mathcal{E}_j] \cdot 2 \cdot \mathbb{E}[\operatorname{dist}(v, \widehat{\textsc{Center}}_{l+i}(v))/2^{l+i} \mid \neg\mathcal{E}_{l+i}] \\
&\leq \sum_{0<l<L} \sum_{0\leq i\leq L-l} 2^{-i} \cdot 2 \cdot \mathbb{E}[\operatorname{dist}(v, \widehat{\textsc{Center}}_{l+i}(v))/2^{l+i} \mid \neg\mathcal{E}_{l+i}] \\
&< \sum_{0<l<L} \sum_{0\leq i\leq \infty} 2^{-i} \cdot 2 \cdot \mathbb{E}[\operatorname{dist}(v, \widehat{\textsc{Center}}_{l}(v))/2^{l} \mid \neg\mathcal{E}_{l}] \\
&= O(1) \cdot \mathbb{E}\left[\sum_{0\leq l\leq L, \widehat{\mathcal{C}}_l(v)\neq\emptyset} \operatorname{dist}(v, \widehat{\textsc{Center}}_{l}(v))/2^{l}\right] \\
&= O(\log n).
\end{aligned}$$

Where we use that the events $\mathcal{E}_j$ are independent as levels are independent from each other, and so are events $\mathcal{E}_j$ from distances-to-centers for level $\neq j$. We then use that $\mathcal{E}_j$ occurs with probability at most $1/2$ by Lemma 4.1, observe a geometric sum, and appeal to the bound on the distances to centers in $\widehat{C}_l$ from the last paragraph.

Correctness of the second property. We fix an arbitrary level $0 < l < L$, and vertices $v, w \in V$, and we want to show $\mathbb{P}[\widehat{\mathcal{C}}_l(w) \neq \widehat{\mathcal{C}}_l(v)] = O\left(\frac{\operatorname{dist}_G(w,v)}{2^l}\right)$. We prove this property by a simple case distinction. We use $\Delta = \operatorname{dist}_G(v, w)$.

- if $\widetilde{C}(w) \neq \widetilde{C}(v)$: Consider the call to Algorithm 3 in the for-loop iteration for cluster $\widetilde{C}^u$ with $w \in \widetilde{C}^u$. The sub-procedure initializes $\hat{X} \leftarrow V \setminus \widetilde{C}(w)$. Since by assumption $v \notin \widetilde{C}^u$, initially $\operatorname{dist}_G(\hat{X}, u) \leq \operatorname{dist}_G(u, w)$. It chooses $r^0 \geq 2\Delta$ with probability $1 - \frac{128\cdot\Delta}{2^l}$. Since it adds $B(\hat{X}, r^0/2) \subseteq B_{T^0}(\hat{X}, r^0)$ to $\hat{X}$ (we use $\frac{1}{1+\epsilon} \leq 2$), the above event results in v, w being both added to $\hat{X}$.

  Since $\hat{X}$ is only increasing in later steps, and $\widehat{C}^w = \widetilde{C}^w \setminus \hat{X}$ for the final set $\hat{X}$, we have that w is not clustered with probability $1 - \Omega(\Delta/2^l)$. The same argument applies to v by symmetry, and thus, the probability that v, w are both not clustered is $1 - \Omega(\Delta/2^l)$. Thus, the u and v are only clustered and separated with probability $O(\Delta/2^l)$.

- otherwise (if $\widetilde{C}(w) = \widetilde{C}(v)$): in this case, the property follows immediately from the following lemma due to [BEL20]. We point out that the precise statement in [BEL20] only bounds the probability of endpoints of edges being separated, however, the statement below is implicit.

  Lemma 4.2 (see Lemma 3.1 in [BEL20])**.** *For $n \geq 2$, Algorithm 3 with parameters G, X, D returns a set $\hat{X} \supset X$ such that any vertices $u, v \in V$ are separated by $\hat{X}$ (i.e. one is contained, one is not), with probability $O(\operatorname{dist}(u, v)/D)$.*

5 Tight probabilistic tree embeddings

Tree construction. Let $\mathcal{R}_{\geq 0} = \{\{v\} \mid v \in V\}, \mathcal{R}_1, \dots, \mathcal{R}_{\geq L} = \{V\}$ be a *refining hierarchical (approximate) random-shift decomposition* with the properties described in Remark 1.2 as obtained in Section 3 and Section 4. For convenience, we extend the associated functions $\textsc{Center}_{\geq l}$ to yield for each cluster $C \in \mathcal{R}_{\geq l}$ the center $\textsc{Center}_{\geq l}(C)$ (since all vertices in C agree on the same center, the image consists only of a single vertex). Now, consider the tree T that has a node associated with each cluster $C \in \mathcal{R}_{\geq l}$, and for $l < L$, an edge from the node to the node associated with $C \subseteq C' \in \mathcal{R}_{\geq l+1}$ of weight $dist_G(\textsc{Center}_{\geq l}(C), \textsc{Center}_{\geq l+1}(C'))$.

Correctness. We next show that T is an $O(\log n)$-approximate probabilistic tree embedding: fix any pair of vertices $u, v \in V$. Let $0 < l \leq L$ be the smallest integer, such that u and v end up in the same cluster C in $\mathcal{R}_{\geq l}(v)$, i.e. $\mathcal{R}_{\geq l}(u) = \mathcal{R}_{\geq l}(v)$. Note that because the partitions are refining, u and v must also be in the same cluster at every higher level.

Then, it is not hard to see that the distance from u to v consists of the weight of the paths from u to the center $\textsc{Center}_{\geq l}(u) = \textsc{Center}_{\geq l}(v)$ and then to v in T. It is thus not hard to see that we can write the expected distance from u to v in T as

$$\begin{aligned}
\mathbb{E}[\text{dist}_T(u,v)] &= \sum_{0 \leq l < L} \mathbb{P}[\mathcal{R}_{\geq l}(u) \neq \mathcal{R}_{\geq l}(v)] \cdot \left(\sum_{z \in \{u,v\}} \text{dist}_G(\textsc{Center}_{\geq l}(z), \textsc{Center}_{\geq l+1}(z)) \right) \\
&\leq 2 \cdot \sum_{0 \leq l < L} \mathbb{P}[\mathcal{R}_{\geq l}(u) \neq \mathcal{R}_{\geq l}(v)] \cdot \left(\sum_{z \in \{u,v\}} \text{dist}_G(z, \textsc{Center}_{\geq l+1}(z)) \right) \\
&\leq 2 \cdot \sum_{0 \leq l < L} O(\text{dist}_G(u,v)/2^l) \cdot \left(\sum_{z \in \{u,v\}} \text{dist}_G(z, \textsc{Center}_{\geq l+1}(z)) \right) \\
&\leq O(\text{dist}_G(u,v)) \cdot \sum_{0 \leq l < L} \left(\sum_{z \in \{u,v\}} \frac{\text{dist}_G(z, \textsc{Center}_{\geq l+1}(z))}{2^l} \right) \\
&= O(\text{dist}_G(u,v)) \cdot O(\log n).
\end{aligned}$$

where we use in the first inequality that by the triangle inequality

$$\text{dist}_G(\textsc{Center}_{\geq i}(z), \textsc{Center}_{\geq i+1}(z)) \leq \text{dist}_G(\textsc{Center}_{\geq i}(z), z) + \text{dist}_G(z, \textsc{Center}_{\geq i+1}(z))$$

and that $\text{dist}_G(\textsc{Center}_{\geq 0}(z), z) = 0$ because $\textsc{Center}_{\geq 0}(z) = z$; in the second inequality, we use the second property of Remark 1.2 (or rather Theorem 1.1); in the third, we re-arrange terms, and in the final inequality, we exchange sums and use the first property of Remark 1.2 for each $z \in \{u, v\}$ separately.

Note that in the construction of T, we used access to the exact distance between cluster centers. While these distances are readily available from the computations in Algorithm 1, when instead using the approximate Algorithm 2, we only have access to the *approximate* distances. However, approximate distances can only increase the stretch by a constant factor, so the asymptotic guarantee remains unchanged.

Implementation. Our analysis above, and inspecting Algorithm 1 straightforwardly allows us to conclude that in the sequential model of computation, only $O(\log n)$ SSSP computations on an $O(m)$ edge graph suffice and that the additional time to construct T is at most $O(m \log n)$. Using the linear-time SSSP algorithm by Thorup [Tho99], we thus obtain runtime $O(m \log n)$. This yields the claim for the sequential setting in Theorem 1.3.

In the parallel, distributed, and semi-streaming models, we refer the reader to [BEL20] for details on how to implement Algorithm 2 and tree T efficiently. We merely point out, that since the publication of [BEL20], the state-of-the-art to compute approximate SSSP in parallel has been significantly advanced (see [Li20, RGH+22]) which speeds-up their parallel implementation by exchanging the result from [RGH+22] with their parallel approximate SSSP implementation. [RGH+22] also de-randomized the previously fastest SSSP algorithm in the CONGEST model. This yields the remaining claims in Theorem 1.3.

6 Tight ℓ_1-Oblivious Routing

Our description of the construction of the ℓ_1-oblivious routing follows closely [REGH22] (we slightly modify their construction to be simpler and yield tight approximation). Our analysis, however, deviates significantly from [REGH22], and is perhaps closer to the approach in [ZGY+22]. In the construction below, we require an additional SSSP subroutine that returns distance estimates $\widehat{\text{dist}}(s,t)$ for all $t \in V$ such that for every edge $(u,v) \in E$, $|\widehat{\text{dist}}(s,u) - \widehat{\text{dist}}(s,v)| \leq 2w(u,v)$.

Construction of the Oblivious Routing. For efficiency reasons, we compute the ℓ_1-oblivious routing $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ as a product of two matrices $\boldsymbol{B} \in \mathbb{R}^{m \times |\mathcal{P}|}$ and $\boldsymbol{C} \in \mathbb{R}^{|\mathcal{P}| \times n}$ where $\mathcal{P}$ is a path collection. The idea is that in $\boldsymbol{C}$, we can add an entire path from $\mathcal{P}$ to the column of any vertex v in $\boldsymbol{C}$ and thus require only $O(\log n)$ bits instead of $O(|P| \log n)$ bits. For $\boldsymbol{B}$, we have to store for each path $P \in \mathcal{P}$ the edge entries explicitly and thus need $O(|P| \log n)$ bits, and thus a total of $O(\sum_{P \in \mathcal{P}} |P| \log n)$ bits. This makes it cheap to use paths $P \in \mathcal{P}$ multiple times across different columns of $\boldsymbol{C}$, even if P consists of many edges.

While it is not possible to efficiently compute $\boldsymbol{A}$ explicitly, this yields an efficient way to apply $\boldsymbol{A}$, meaning that for any demand $\boldsymbol{d}$, we can compute $\boldsymbol{A}\boldsymbol{d}$ efficiently by first computing $\boldsymbol{g} = \boldsymbol{C}\boldsymbol{d}$ and then $\boldsymbol{f} = \boldsymbol{B}\boldsymbol{g}$. Similarly, we can also compute products $\boldsymbol{A}^T\boldsymbol{v}$ efficiently.

Oblivious Routing via (approximate) random-shift decompositions. Let us next describe the approximate random-shift decomposition that we use in the construction of $\boldsymbol{B}$ and $\boldsymbol{C}$.

Let $D = 36 \log(n)$. We compute for every $0 < l < L$, and $0 < d \leq D$ an (approximate) random-shift decomposition $\mathcal{C}_{l,d}$ computed from either Algorithm 1 or Algorithm 2 invoked with parameter 2^l. Note that we compute all of these decompositions independently. For each such (sub)partition $\mathcal{C}_{l,d}$, we let $\textsc{Center}_{l,d}$ denote the corresponding center function. For convenience, we define $\textsc{Center}_{0,d}(v) = v$ and $\textsc{Center}_{L,d}(v) = r$ for some arbitrary vertex $r \in V$ and every $v \in V$ and $0 < d \leq D$. In particular, every (approximate) shortest path from a vertex $v \in V$ to one of its centers can be obtained from the concatenation of $O(\log^2 n)$ paths.

Constructing the path collection and $\boldsymbol{B}$. For every (sub)partition $\mathcal{C}_{l,d}$, we let $F_{l,d}$ be the associated forest that was computed by Algorithm 1 or Algorithm 2 and spans the partition sets.

Applying a heavy-light decomposition (see [ST81]), we can decompose $F_{l,d}$ into a collection of edge-disjoint paths $\mathcal{P}_{l,d}$ such that every path in $F_{l,d}$ intersects with $O(\log n)$ such paths only. Finally, for every path $P \in \mathcal{P}_{l,d}$, we add the dyadic sub-segments of P to the collection $\mathcal{P}$ that is for every $0 \leq i \leq \lceil \lg(n) \rceil$ and $j \in (0, \lceil |P|/2^i \rceil)$, we add the segment in P from the $(j \cdot 2^i + 1)$-th to the $(\min\{(j+1) \cdot 2^i, |P|\})$-th vertex to $\mathcal{P}$.

Constructing the routing matrix $\boldsymbol{C}$. To construct $\boldsymbol{C}$, for every vertex $v \in V$, $0 \leq l \leq L$, we define for every cluster $C \in \mathcal{C}_{l,d}$ for $0 < d \leq D$, $p_{l,d,C}(v) = \min\{1, \widehat{\text{dist}}(v, V \setminus C)/2^l\}$ (note that if $v \notin C$, then $p_{l,d,C}(v) = 0$). We let $w_l(v) = \sum_{0<d\leq D} \sum_{C \in \mathcal{C}_{l,d}} p_{l,d,C}(v)$.

We now construct $\boldsymbol{C}$ such that the column for $v \in V$ routes $f_{l,d,d',C,C'}(v) = \frac{p_{l,d,C}(v)}{w_l(v)} \cdot \frac{p_{l+1,d',C'}(v)}{w_{l+1}(v)}$ units of flow along some $\textsc{Center}_{l,d}(v)$ to $\textsc{Center}_{l+1,d'}(v)$ path, for every $d, d' \in [D]$ and $C \in \mathcal{C}_{l,d}, C' \in \mathcal{C}_{l+1,d'}$. We take the path between the centers to be the (approximate) shortest paths via some vertex $w \in V$ that also needs to route flow between them, i.e. we let

$$m_{l,d,d',C,C'} = \min_{w \in V, f_{l,d,d',C,C'}(w) \neq 0} \text{dist}_{T_{l,d}}(\textsc{Center}_{l,d}(w), w) + \text{dist}_{T_{l+1,d'}}(w, \textsc{Center}_{l+1,d'}(w))$$

and let $P_{l,d,d',C,C'}$ be the corresponding $\textsc{Center}_{l,d}(v)$ to $\textsc{Center}_{l+1,d'}(v)$ path which can be obtained from the composition of $O(\log^2 n)$ paths in $\mathcal{P}$. For each such segment, we add $f_{l,d,d',C,C'}(v)$ to its entry. This completes the construction of routing matrix $\boldsymbol{C}$ and thus of our oblivious routing matrix $\boldsymbol{A}$.

Correctness. We start by establishing that $\boldsymbol{A}$ indeed is an oblivious routing, i.e. for every demand $\boldsymbol{d}$, flow $\boldsymbol{f} = \boldsymbol{A}\boldsymbol{d}$ routes the demand $\boldsymbol{d}$.

To this end, let us first analyze $\boldsymbol{g} = \boldsymbol{A}\mathbf{1}_v$ for some vertex $v \in V$. We claim that $\boldsymbol{g}$ is a distribution over vr-paths (recall that r is the arbitrary vertex that is chosen to be the center of all vertices at level L). This yields overall correctness since it implies that $\boldsymbol{A}\boldsymbol{d}(v)$ sends $\boldsymbol{d}(u)$ units of flow away from u, conserves flow at other vertices, and sends $\boldsymbol{d}(u)$ units of flow into r. Since $\sum_v \boldsymbol{d}(v) = 0$ as $\boldsymbol{d}$ is a demand, the positive and negative flows cancel the flow into r (except for $-\boldsymbol{d}(r)$) and the result follows.

To see that $\boldsymbol{g} = \boldsymbol{A}\mathbf{1}_v$ is a distribution over vr-paths, observe that for level $0 \leq l < L$, we have for every $0 < d \leq D$, $C \in \mathcal{C}_{l,d}$ that the out-flow of $\textsc{Center}_{l,d}(v)$ is

$$\begin{aligned}\sum_{0<d'\leq D, C' \in \mathcal{C}_{l+1,d'}} f_{l,d,d',C,C'}(v) &= \sum_{0<d'\leq D, C' \in \mathcal{C}_{l+1,d'}} \frac{p_{l,d,C}(v)}{w_l(v)} \cdot \frac{p_{l+1,d',C'}(v)}{w_{l+1}(v)} \\ &= \frac{p_{l,d,C}(v)}{w_l(v)} \cdot \sum_{0<d'\leq D, C' \in \mathcal{C}_{l+1,d'}} \frac{p_{l+1,d',C'}(v)}{w_{l+1}(v)} = \frac{p_{l,d,C}(v)}{w_l(v)}\end{aligned}$$

For $0 < l \leq L$, we further have from a similar calculation that the in-flow is $\frac{p_{l,d,C}(v)}{w_l(v)}$ and thus the flow is conserved at each center on level $0 < l < L$. This immediately yields the claim.

Tight Competitive Ratio. Before we dive into the proof of the competitive ratio, let us first observe that for every vertex $v \in V$, level $0 < l < L$ and $0 < d \leq D$, and either by an argument via Lemma 3.3 or directly by Lemma 4.1, we have that with probability at least $1/2$, v is clustered and

$B(v, 2^l/8) \subseteq \mathcal{C}_{l,d}(v)$. Thus, $\mathbb{E}[p_{v,l,d}(v)] \geq 1/8$. It follows immediately from a Chernoff bound that $D/16 \leq w_l(v) \leq D$ with probability at least $1 - n^{-3}$. We condition on this event and the event that if v is clustered by $\mathcal{C}_{l,d}$, then the distance to its center is bounded, i.e. $\text{dist}(v, \textsc{Center}_{l,d}(v)) \leq 2^l \cdot 10 \log n$, for all choices of $v \in V$, $0 < l < L$ and $0 < d \leq D$. By a union bound, these events occur with probability at least $1 - O(n^{-2})$.

Let us now bound the competitive ratio of $\boldsymbol{A}$. From matrix norms, we have that the competitive ratio is equivalent to $\max_{(u,v)\in E} \frac{\|\boldsymbol{A}(\mathbf{1}_u - \mathbf{1}_v)\|_1}{\text{dist}(u,v)}$ (see for example [FKGS24]). For the rest of the proof, let us fix $u, v \in V$ so that they achieve the maximum ratio among all vertex pairs. We observe that

$$\|\boldsymbol{A}(\mathbf{1}_u - \mathbf{1}_v)\|_1 \leq \sum_{0\leq l<L} \sum_{\substack{0<d\leq D, C\in\mathcal{C}_{l,d},\\ 0<d'\leq D, C'\in\mathcal{C}_{l+1,d'}}} |f_{l,d,d',C,C'}(v) - f_{l,d,d',C,C'}(u)| \cdot w(P_{l,d,d',C,C'}) \tag{7}$$

since each term $|f_{l,d,d',C,C'}(v) - f_{l,d,d',C,C'}(u)|$ yields the amount of flow sent between center C and C' along path $P_{l,d,d',C,C'}$ (from the demand, v uses the path in the forward direction, u in the backwards direction) and we are summing over all combinations of clusters at consecutive levels, and levels. Note that the inequality is due only to further cancellations, i.e. an edge $e \in E$ might occur on two (or more) paths $P_{l,d,d',C,C'}$ and $P_{l',d'',d''',C'',C'''}$ and the flows along these paths goes into the opposite directions, thus yielding a cancellation on e.

Let us define for each $0 \leq l < L$, the variable

$$Y_l = \sum_{\substack{0<d\leq D, C\in\mathcal{C}_{l,d},\\ 0<d'\leq D, C'\in\mathcal{C}_{l+1,d'}}} |f_{l,d,d',C,C'}(v) - f_{l,d,d',C,C'}(u)| \cdot w(P_{l,d,d',C,C'}).$$

The goal of this section is to argue about the sum $Y = \sum_{0\leq l<L} Y_l$ and bound the sum by $O(\log n)$ w.h.p. The key to obtaining this bound is to find an upper bound on the expectation of Y_l, i.e., $\mathbb{E}[Y_l]$. This allows us to conclude the result by a simple Chernoff bound.

To avoid working with the absolute value when analyzing Y_l, we observe

$$\begin{aligned} &|f_{l,d,d',C,C'}(v) - f_{l,d,d',C,C'}(u)| \\ &= \max\{0, f_{l,d,d',C,C'}(v) - f_{l,d,d',C,C'}(u)\} + \max\{0, f_{l,d,d',C,C'}(u) - f_{l,d,d',C,C'}(v)\} \end{aligned} \tag{8}$$

We next analyze the non-canceled flow sent between a specific pair of level l and $l+1$ centers more carefully. For any $0 < d \leq D, C \in \mathcal{C}_{l,d}, 0 < d' \leq D, C' \in \mathcal{C}_{l+1,d'}$

$$\begin{aligned} f_{l,d,d',C,C'}(v) - f_{l,d,d',C,C'}(u) &= \frac{p_{l,d,C}(v)}{w_l(v)} \cdot \frac{p_{l+1,d',C'}(v)}{w_{l+1}(v)} - \frac{p_{l,d,C}(u)}{w_l(u)} \cdot \frac{p_{l+1,d',C'}(u)}{w_{l+1}(u)} \\ &= O\left(\frac{p_{l,d,C}(v)p_{l+1,d',C'}(v) - p_{l,d,C}(u)p_{l+1,d',C'}(u)}{D^2}\right). \end{aligned} \tag{9}$$

The key observation now is that $\text{dist}(u, V \setminus C) \leq \text{dist}(v, V \setminus C) - \Delta$ for $\Delta = d(u, v)$ by the triangle inequality. This yields

$$\begin{aligned} \text{dist}(v, V \setminus C) \cdot \text{dist}(v, V \setminus C') - &\text{dist}(u, V \setminus C) \cdot \text{dist}(u, V \setminus C') \\ &\leq \text{dist}(v, V \setminus C) \cdot \text{dist}(v, V \setminus C') - (\text{dist}(v, V \setminus C) - \Delta) \cdot (\text{dist}(v, V \setminus C') - \Delta) \\ &= \Delta(\text{dist}(v, V \setminus C) + \text{dist}(v, V \setminus C')) - \Delta^2 \\ &< \Delta(\text{dist}(v, V \setminus C) + \text{dist}(v, V \setminus C'). \end{aligned}$$

Note that by definition of $p_{l,d,C}(v)$ (and our assumption on $\widehat{\text{dist}}$), this implies

$$p_{l,d,C}(v)p_{l+1,d',C'}(v) - p_{l,d,C}(u)p_{l+1,d',C'}(u) \leq \frac{2\Delta}{2^l} \cdot (p_{l,d,C}(v) + p_{l+1,d',C'}(v))$$

since capping the values $p_{l,d,C}(v), p_{l+1,d',C'}(v), p_{l,d,C}(u), p_{l+1,d',C'}(u)$ at 1 can only tighten the inequality. We conclude that

$$\begin{aligned}
&\sum_{\substack{0<d\leq D, C\in\mathcal{C}_{l,d},\\ 0<d'\leq D, C'\in\mathcal{C}_{l+1,d'}}} |f_{l,d,d',C,C'}(v) - f_{l,d,d',C,C'}(u)| \\
&= O\left(\frac{\Delta}{2^l D^2} \cdot \sum_{\substack{0<d\leq D, C\in\mathcal{C}_{l,d},\\ 0<d'\leq D, C'\in\mathcal{C}_{l+1,d'}}} \sum_{z\in\{u,v\}} (p_{l,d,C}(z) + p_{l+1,d',C'}(z))\right). \qquad (10)\\
&= O\left(\frac{\Delta}{2^l D^2} \cdot D \cdot (w_l(v) + w_{l+1}(v) + w_l(u) + w_{l+1}(u))\right) \\
&= O(\Delta/2^l)
\end{aligned}$$

Returning to our analysis of Y_l, we obtain that for $\Gamma_{l,d}(z) = \text{dist}(z, \textsc{Center}_{l,d}(z))$, we can conclude that $\mathbb{E}[Y_l]$ is bound from above by

$$\begin{aligned}
&\mathbb{E}\left[\sum_{\substack{0<d\leq D, C\in\mathcal{C}_{l,d},\\ 0<d'\leq D, C'\in\mathcal{C}_{l+1,d'}}} |f_{l,d,d',C,C'}(u) - f_{l,d,d',C,C'}(v)| \cdot \left(\Gamma_{l,d}(z) + \Gamma_{l+1,d'}(z)\right)\right] \qquad (11)\\
&= O(\Delta) \cdot (r_l(v) + r_{l+1}(v) + r_l(u) + r_{l+1}(u))
\end{aligned}$$

where we let $r_l(v), r_{l+1}(v), r_l(u), r_{l+1}(u)$ be as in the ball growing process described in Section 3 for level $l/\ l+1$ and vertex $u/\ v$, respectively.

We use next that the inequalities (10) are deterministic and so each variable Y_l has value at most

$$O(\Delta) \cdot (r_l(v) + X_l(v) + r_{l+1}(v) + X_{l+1}(v) + r_l(u) + X_l(u) + r_{l+1}(u) + X_{l+1}(u))$$

for $X_l(v), X_{l+1}(v), X_l(u), X_{l+1}(u) \sim \textsc{Exp}(1)$ by the fact that distances-to-centers are stochastically dominated by an r_l term and an exponentially distributed variable $X_l \sim \textsc{Exp}(1)$ as explicitly pointed out both in Section 3 and Section 4.

We have from the Chernoff bound for exponentially-distributed random variables that the sums $X(u) = \sum_{0\leq l\leq L} X_l(u)$ and $X(v) = \sum_{0\leq l\leq L} X_l(v)$ are both upper bounded by $10\log n$ with proba-

bility at least $1 - O(n^{-4})$. Conditioning on this event finally yields

$$\begin{aligned}\sum_{0\le l<L} Y_l &= O\left(\sum_{0\le l<L} \Delta\cdot(r_l(v)+X_l(v)+r_{l+1}(v)+X_{l+1}(v)+r_l(u)+X_l(u)+r_{l+1}(u)+X_{l+1}(u))\right)\\ &= O\left(\sum_{0\le l\le L} \Delta\cdot(r_l(v)+X_l(v)+r_l(u)+X_l(u))\right)\\ &= O\left(\Delta\cdot\left(\left(\sum_{0\le l\le L} r_l(v)\right)+\left(\sum_{0\le l\le L} r_l(u)\right)+\left(\sum_{0\le l\le L} X_l(u)\right)+\left(\sum_{0\le l\le L} X_l(v)\right)\right)\right)\\ &= O(\Delta\log n)\end{aligned}$$

where in the last inequality, we use Lemma 3.2 and our previous conditioning. This concludes the proof.

Implementation. For the implementation, we refer the reader to Section 5 for the implementation of (approximate) random-shift decomposition. The remaining details can be found in [REGH22], where a fast SSSP routine is given that computes the distance function $\widehat{\text{dist}}$ as required. We note that [REGH22] does not compute the quantities $m_{l,d,d',C,C'}$ since they only route between clusters where $C \subseteq C'$, and thus, the shortest path between centers is already pre-computed. However, each vertex $v \in V$ participates in at most $L \cdot D^2$ such minimization problems; thus, these quantities can be implemented efficiently in their minor-aggregation model of computation, which in turn is efficiently implementable in PRAM and CONGEST models. Analogously, paths $P_{l,d,d',C,C'}$ can be computed efficiently. Theorem 1.4 follows.

We finally note that our proof can be turned into a verifier of whether $\boldsymbol{A}$ is of good quality: we can simply check $\boldsymbol{A}(\mathbf{1}_u - \mathbf{1}_v)$ for all edges $(u, v) \in E$. Using this approach, we can further convert our algorithm from a Monte-Carlo algorithm into a Las-Vegas algorithm.

Acknowledgment

We thank the FOCS reviewers for their careful and insightful feedback. We are especially grateful to Bernhard Haeupler and Arnold Filtser for clarifying the history of the problem and for pointing out the overlap with [HW16, CD21].

References

[AA97] Baruch Awerbuch and Yossi Azar. Buy-at-bulk network design. In *Proceedings 38th Annual Symposium on Foundations of Computer Science*, pages 542–547. IEEE, 1997. 1, 23

[ABN08] Ittai Abraham, Yair Bartal, and Ofer Neiman. Nearly tight low stretch spanning trees. In *2008 49th Annual IEEE Symposium on Foundations of Computer Science*, pages 781–790. IEEE, 2008. 23

[ACB17] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR, 2017. 1

[ACE+20] Ittai Abraham, Shiri Chechik, Michael Elkin, Arnold Filtser, and Ofer Neiman. Ramsey spanning trees and their applications. *ACM Transactions on Algorithms (TALG)*, 16(2):1–21, 2020. 24

[AKPW95] Noga Alon, Richard M Karp, David Peleg, and Douglas West. A graph-theoretic game and its application to the k-server problem. *SIAM Journal on Computing*, 24(1):78–100, 1995. 1

[AN19] Ittai Abraham and Ofer Neiman. Using petal-decompositions to build a low stretch spanning tree. *SIAM Journal on Computing*, 48(2):227–248, 2019. 23, 24

[ASZ20] Alexandr Andoni, Clifford Stein, and Peilin Zhong. Parallel approximate undirected shortest paths via low hop emulators. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 322–335, 2020. 24

[Bar96] Yair Bartal. Probabilistic approximation of metric spaces and its algorithmic applications. In *Proceedings of 37th Conference on Foundations of Computer Science*, pages 184–193. IEEE, 1996. 1, 2

[Bar98] Yair Bartal. On approximating arbitrary metrices by tree metrics. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 161–168, 1998. 1

[Bar04] Yair Bartal. Graph decomposition lemmas and their role in metric embedding methods. In *European Symposium on Algorithms*, pages 89–97. Springer, 2004. 1

[BEL20] Ruben Becker, Yuval Emek, and Christoph Lenzen. Low diameter graph decompositions by approximate distance computation. In *11th Innovations in Theoretical Computer Science Conference (ITCS 2020)*, pages 50–1. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2020. 3, 4, 5, 9, 12, 14, 24

[BGS17] Guy E Blelloch, Yan Gu, and Yihan Sun. Efficient construction of probabilistic tree embeddings. In *44th International Colloquium on Automata, Languages, and Programming (ICALP 2017)*, pages 26–1. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2017. 4, 24

[BGSS20] Guy E Blelloch, Yan Gu, Julian Shun, and Yihan Sun. Parallelism in randomized incremental algorithms. *Journal of the ACM (JACM)*, 67(5):1–27, 2020. 4, 24

[BGT12] Guy E Blelloch, Anupam Gupta, and Kanat Tangwongsan. Parallel probabilistic tree embeddings, k-median, and buy-at-bulk network design. In *Proceedings of the twenty-fourth annual ACM symposium on Parallelism in algorithms and architectures*, pages 205–213, 2012. 24

[BGWN20] Aaron Bernstein, Maximilian Probst Gutenberg, and Christian Wulff-Nilsen. Near-optimal decremental sssp in dense weighted digraphs. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1112–1122. IEEE, 2020. 24

[BKE+24] Sabyasachi Basu, Nadia Kōshima, Talya Eden, Omri Ben-Eliezer, and C Seshadhri. A sublinear algorithm for approximate shortest paths in large networks. *arXiv preprint arXiv:2406.08624*, 2024. 1

[CD21] Artur Czumaj and Peter Davies. Exploiting spontaneous transmissions for broadcasting and leader election in radio networks. *Journal of the ACM (JACM)*, 68(2):1–22, 2021. 3, 18

[Che14] Shiri Chechik. Approximate distance oracles with constant query time. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 654–663, 2014. 24

[Che15] Shiri Chechik. Approximate distance oracles with improved bounds. In *Proceedings of the forty-seventh annual ACM symposium on Theory of Computing*, pages 1–10, 2015. 24

[CKNZ01] Chandra Chekuri, Sanjeev Khanna, Joseph Naor, and Leonid Zosin. Approximation algorithms for the metric labeling problem via a new linear programming formulation. In *Proceedings of the 12th Annual ACM-SIAM Symposium on Discrete Algorithms*, Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms, pages 109–118, 2001. 2001 Operating Section Proceedings, American Gas Association ; Conference date: 30-04-2001 Through 01-05-2001. 1

[CTW24] Samantha Chen, Puoya Tabaghi, and Yusu Wang. Learning ultrametric trees for optimal transport regression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 20657–20665, 2024. 1

[CZ20] Shiri Chechik and Tianyi Zhang. Dynamic low-stretch spanning trees in subpolynomial time. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 463–475. SIAM, 2020. 24

[DGPW14] Daniel Delling, Andrew V Goldberg, Thomas Pajor, and Renato F Werneck. Robust distance queries on massive networks. In *Algorithms-ESA 2014: 22th Annual European Symposium, Wroclaw, Poland, September 8-10, 2014. Proceedings 21*, pages 321–333. Springer, 2014. 1

[DSHK+11] Atish Das Sarma, Stephan Holzer, Liah Kor, Amos Korman, Danupon Nanongkai, Gopal Pandurangan, David Peleg, and Roger Wattenhofer. Distributed verification and hardness of distributed approximation. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 363–372, 2011. 4, 24

[EEST05] Michael Elkin, Yuval Emek, Daniel A Spielman, and Shang-Hua Teng. Lower-stretch spanning trees. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 494–503, 2005. 23

[EN18] Michael Elkin and Ofer Neiman. Efficient algorithms for constructing very sparse spanners and emulators. *ACM Transactions on Algorithms (TALG)*, 15(1):1–29, 2018. 24

[ER09] Matthias Englert and Harald Räcke. Oblivious routing for the lp-norm. In *2009 50th Annual IEEE Symposium on Foundations of Computer Science*, pages 32–40. IEEE, 2009. 2

[FG19] Sebastian Forster and Gramoz Goranci. Dynamic low-stretch trees via dynamic low-diameter decompositions. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 377–388, 2019. 24

[FGdV22] Sebastian Forster, Martin Grösbacher, and Tijn de Vos. An improved random shift algorithm for spanners and low diameter decompositions. In *25th International Conference on Principles of Distributed Systems (OPODIS 2021)*, pages 16–1. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2022. 24

[FGH21] Sebastian Forster, Gramoz Goranci, and Monika Henzinger. Dynamic maintenance of low-stretch probabilistic tree embeddings with applications. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1226–1245. SIAM, 2021. 24

[FKGS24] Cella Florescu, Rasmus Kyng, Maximilian Probst Gutenberg, and Sushant Sachdeva. Optimal electrical oblivious routing on expanders. In *51st International Colloquium on Automata, Languages, and Programming (ICALP 2024)*, pages 65–1. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2024. 16

[FL18] Stephan Friedrichs and Christoph Lenzen. Parallel metric tree embedding based on an algebraic view on moore-bellman-ford. *Journal of the ACM (JACM)*, 65(6):1–55, 2018. 4, 24

[Fox24] Emily Fox. A simple deterministic near-linear time approximation scheme for transshipment with arbitrary positive edge costs. In *32nd Annual European Symposium on Algorithms (ESA 2024)*, pages 56–1. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2024. 1, 24

[FRT03] Jittat Fakcharoenphol, Satish Rao, and Kunal Talwar. A tight bound on approximating arbitrary metrics by tree metrics. In *Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing*, STOC '03, page 448–455, New York, NY, USA, 2003. Association for Computing Machinery. 1, 2, 24

[GL14] Mohsen Ghaffari and Christoph Lenzen. Near-optimal distributed tree embedding. In *International Symposium on Distributed Computing*, pages 197–211. Springer, 2014. 4, 24

[HW16] Bernhard Haeupler and David Wajc. A faster distributed radio broadcast primitive. In *Proceedings of the 2016 ACM Symposium on Principles of Distributed Computing*, pages 361–370, 2016. 3, 18

[KKM+08] Maleq Khan, Fabian Kuhn, Dahlia Malkhi, Gopal Pandurangan, and Kunal Talwar. Efficient distributed approximation algorithms via probabilistic tree embeddings. In *Proceedings of the twenty-seventh ACM symposium on Principles of distributed computing*, pages 263–272, 2008. 24

[KT02] Jon Kleinberg and Eva Tardos. Approximation algorithms for classification problems with pairwise relationships: Metric labeling and markov random fields. *Journal of the ACM (JACM)*, 49(5):616–639, 2002. 1

[LCCS18] Hugo Lavenant, Sebastian Claici, Edward Chien, and Justin Solomon. Dynamical optimal transport on discrete surfaces. *ACM Transactions on Graphics (TOG)*, 37(6):1–16, 2018. 1

[Li20] Jason Li. Faster parallel algorithm for approximate shortest path. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 308–321, 2020. 1, 14, 24

[LR18] Jaeho Lee and Maxim Raginsky. Minimax statistical learning with wasserstein distances. *Advances in Neural Information Processing Systems*, 31, 2018. 1

[MN07] Manor Mendel and Assaf Naor. Ramsey partitions and proximity data structures. *Journal of the European Mathematical Society*, 9(2):253–275, 2007. 24

[MPX13] Gary L. Miller, Richard Peng, and Shen Chen Xu. Parallel graph decompositions using random shifts. In *Proceedings of the Twenty-Fifth Annual ACM Symposium on Parallelism in Algorithms and Architectures*, SPAA '13, page 196–203, New York, NY, USA, 2013. Association for Computing Machinery. 2, 3, 8, 24

[MS09] Manor Mendel and Chaya Schwob. Fast ckr partitions of sparse graphs. *Chicago Journal OF Theoretical Computer Science*, 2:1–18, 2009. 24

[NT12] Assaf Naor and Terence Tao. Scale-oblivious metric fragmentation and the nonlinear dvoretzky theorem. *Israel Journal of Mathematics*, 192:489–504, 2012. 24

[PSM+11] Georgios A Pavlopoulos, Maria Secrier, Charalampos N Moschopoulos, Theodoros G Soldatos, Sophia Kossida, Jan Aerts, Reinhard Schneider, and Pantelis G Bagos. Using graph theory to analyze biological networks. *BioData mining*, 4:1–27, 2011. 1

[Räc08] Harald Räcke. Optimal hierarchical decompositions for congestion minimization in networks. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 255–264, 2008. 1

[REGH22] Václav Rozhoň, Michael Elkin, Christoph Grunau, and Bernhard Haeupler. Deterministic low-diameter decompositions for weighted graphs and distributed and parallel applications. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1114–1121. IEEE, 2022. 1, 4, 5, 14, 18, 24

[RGH+22] Václav Rozhoň, Christoph Grunau, Bernhard Haeupler, Goran Zuzic, and Jason Li. Undirected $(1+\epsilon)$-shortest paths via minor-aggregates: near-optimal deterministic parallel and distributed algorithms. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, pages 478–487, 2022. 14, 24

[RR98] Yuri Rabinovich and Ran Raz. Lower bounds on the distortion of embedding finite metric spaces in graphs. *Discrete & Computational Geometry*, 19:79–94, 1998. 1

[She17] Jonah Sherman. Generalized preconditioning and undirected minimum-cost flow. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 772–780. SIAM, 2017. 1

[SST+19] Geoffrey Schiebinger, Jian Shu, Marcin Tabaka, Brian Cleary, Vidya Subramanian, Aryeh Solomon, Joshua Gould, Siyan Liu, Stacie Lin, Peter Berube, et al. Optimal-transport analysis of single-cell gene expression identifies developmental trajectories in reprogramming. *Cell*, 176(4):928–943, 2019. 1

[ST81] Daniel D Sleator and Robert Endre Tarjan. A data structure for dynamic trees. In *Proceedings of the thirteenth annual ACM symposium on Theory of computing*, pages 114–122, 1981. 15

[Tho99] Mikkel Thorup. Undirected single-source shortest paths with positive integer weights in linear time. *Journal of the ACM (JACM)*, 46(3):362–394, 1999. 14

[ZGY+22] Goran Zuzic, Gramoz Goranci, Mingquan Ye, Bernhard Haeupler, and Xiaorui Sun. Universally-optimal distributed shortest paths and transshipment via graph-based ℓ_1-oblivious routing. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2549–2579. SIAM, 2022. 1, 14

[Zuz23] Goran Zuzic. A simple boosting framework for transshipment. In *31st Annual European Symposium on Algorithms (ESA 2023)*, pages 104–1. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2023. 1

A Related Work

In this section, we review related work that was not covered in the introduction.

Low-Stretch Spanning Trees (LSSTs). The notion of probabilistic tree embeddings can be strengthened to require sampled trees T to be subgraphs of G, i.e. $T \subseteq G$. The probabilistic tree embeddings are then commonly referred to as *low-stretch spanning trees*. This enables additional applications, most notably, as pre-conditioners to solve Laplacian systems. In fact, the construction in [AA97] is a low-stretch spanning tree. In [EEST05], a construction with polylogarithmic stretch was given, which was subsequently improved to $O(\log n \log\log n)$ [ABN08, AN19].

Fast Algorithms for Tree Embeddings. While [FRT03] already gives an $O(n^2)$ time algorithm for constructing FRT trees, this bound was further improved to $O(m \log^3 n)$ in [MS09] and then to $O(m \log n)$ by [BGS17]. [BGT12, BGSS20, FL18] consider the construction of FRT trees in the parallel setting where the best constructions achieve $\tilde{O}(1)$ depth, and either $O(n^2)$ or $O(m^{1+\delta})$ for some constant $\delta > 0$ work. The best parallel algorithm with $\tilde{O}(1)$ depth and $\tilde{O}(m)$ work achieves approximation $O(\log^2 n)$ [BEL20]. In the distributed CONGEST model, [KKM+08, GL14] yield an algorithm to construct FRT trees in $O(n^{0.5+\delta} + D)$ rounds for any constant $\delta > 0$ where D denotes the unweighted diameter of G. From [DSHK+11], a lower bound of $\Omega(n^{0.5} + D)$ is known.

LSSTs with $O(\log n \log \log n)$ approximation can be computed in time $O(m \log n \log \log n)$ [AN19]. Constructions achieving polylogarithmic approximation running in work $\tilde{O}(m)$ and depth $\tilde{O}(1)$ are known [REGH22].

Fast Algorithms for Tree Embeddings. In a recent line of work, various algorithms achieved $\tilde{O}(1)$ competitive ratio via algorithms with $\tilde{O}(m)$ runtime/work [Li20, ASZ20, RGH+22, Fox24]. Notably, all but the last algorithm can be implemented in parallel with $\tilde{O}(1)$ depth. [RGH+22] can also be implemented in the CONGEST model in $\tilde{O}(\sqrt{n} + D)$ rounds. [RGH+22, Fox24] are deterministic constructions.

Ramsey Trees via the FRT framework. In [MN07], Mendel and Naor showed that the FRT tree algorithm can be shown to yield for any $k \geq 1$, and $u \in V$ that the sampled tree T preserves all distance $\text{dist}_G(u, v)$ for $v \in V$ up to a factor $O(k)$ with probability $n^{-1/k}$. This was achieved by generalizing the proof from [FRT03]. Such trees T are called Ramsey trees and have since received significant further attention. Ramsey trees have deep applications to distance oracles, oblivious routing problems and spanners (see for example [MN07, Che14, Che15]). The trade-off between approximation and success probability has since been reduced significantly [BEL20, NT12, ACE+20]. However, as pointed out in [NT12], it seems unlikely that the proof from [FRT03] can be extended to yield a near-perfect trade-off of $(2k - 1)$ stretch for a given probability.

Random-Shift Decompositions. Random-shift, as proposed by Miller et al.[MPX13], was developed in the context of parallelizing spanner constructions. In [MPX13], for an unweighted graph G and any $k \geq 1$, they give a randomized algorithm to compute a subgraph $H \subseteq G$ with work $O(m)$ and depth $O(k)$ w.h.p. such that H has $O(n^{1+1/k})$ edges and preserves all distances up to an $O(k)$ factor. At a loss of $O(\log k)$, their construction also works for weighted graphs.

Subsequently, Elkin and Neiman [EN18] improved the approximation guarantee to $2k-1$, which is optimal under Erdös' girth conjecture. In [FGdV22], Forster, Grösbacher, Martin and de Vos show that by replacing the exponential distribution with a capped geometric distribution further yields that the number of rounds $O(k)$ can be made deterministic.

Recently, random-shift techniques have inspired dynamic algorithms for probabilistic tree embeddings [FG19, CZ20, FGH21] and shortest-paths problems [BGWN20].

B Chernoff Bound for Sum of Exponentially-Distributed Variables

Given i.i.d. random variables $X_1, X_2, \ldots, X_n \sim \textsc{Exp}(1)$. Let $X = \sum_i X_i$, and $\mathbb{E}[X] = n$, then

$$\mathbb{P}[X > (2+\delta)n] = \mathbb{P}[e^{tX} > e^{t(2+\delta)n}] \leq \frac{\mathbb{E}[e^{tX}]}{e^{t(2+\delta)\mu}} = e^{-t(2+\delta)\mu} \cdot \prod_i \mathbb{E}[e^{tX_i}].$$

for every $t > 0$, using Markov's inequality and then the independence between the random variables.

It remains to use the moment-generating function $M_{X_i}(t) = \mathbb{E}[e^{tX_i}] = \frac{1}{1-t}$ which holds for any $t < 1$. For $t = 1/2$, $M_{X_i}(1/2) = 2$ and by independence $\mathbb{E}[e^{tX_i}] = 2^n$. We thus obtain

$$\mathbb{P}[X > (2+\delta)n] < 2^n \cdot e^{-(2+\delta)n/2} < e^{-\delta n/2}.$$