An Improved Analysis of the Quadtree for High Dimensional EMD

Xi Chen* Rajesh Jayaram† Amit Levi‡ Erik Waingarten§

November 9, 2020

Abstract

The Earth Mover Distance (EMD) between two multi-sets $A, B \subset \mathbb{R}^d$ of size s is the min-cost of bipartite matchings between points in A and B, where cost is measured by distance between points. We study a class of divide-and-conquer algorithms based on a *quadtree* data structure. Our analysis improves on the $O(\min\{\log s, \log d\} \log s)$-approximation of [AIK08, BDI+20], showing that the approximation achieved is in fact $\tilde{\Theta}(\log s)$. As further applications, we give a two-round linear sketch achieving $\tilde{O}(\log s)$-approximation using $\mathrm{polylog}(s, d)$ space, and a compression into a one-round linear sketch at the cost of small additive errors. These sketches result in two-pass and one-pass streaming algorithms, respectively, for approximating EMD. The main conceptual contribution is an analytical framework for studying the quadtree which goes beyond worst-case distortion of randomized tree embeddings.

*Columbia University. xichen@cs.columbia.edu. Supported by NSF IIS-1838154 and NSF CCF-1703925.

†Carnegie Mellon University. rkjayara@cs.cmu.edu. Rajesh Jayaram would like to thank the partial support from the Office of Naval Research (ONR) grant N00014-18-1-2562, and the National Science Foundation (NSF) under Grant No. CCF-1815840.

‡Cheriton School of Computer Science, University of Waterloo. amit.levi@uwaterloo.ca. Research supported by the David R. Cheriton Graduate Scholarship.

§Stanford University and the Simons Institute for Theory of Computing. eaw@cs.columbia.edu. This material is based upon work supported by the National Science Foundation under Award No. 2002201.

Contents

1 Introduction

Let (X, d_X) be a metric space. Given two (multi-)sets $A, B \subseteq X$ of size $|A| = |B| = s$, the Earth Mover distance (EMD) between A and B is

$$\mathrm{EMD}_X(A, B) = \min_{\substack{\text{matching} \\ M \subset A \times B}} \sum_{(a,b) \in M} d_X(a, b).$$

Computational aspects of EMD, and more generally, geometric minimum cost matchings consistently arise in multiple areas of computer science [RTG00, PC19], and its non-geometric version is a classic algorithms topic [Kuh55, Mun57, EK72, AMO93]. Efficient computation of EMD, however, has been a well-known bottleneck in practice [Cut13, GCPB16, AWR17, LYFC19, BDI+20] which motivated many works from the theoretical perspective to study approximation and sketching algorithms for EMD [Cha02, IT03, AIK08, ABIW09, AS14, BI14, ANOY14, She17, KNP19, BDI+20] in both low- and high-dimensional settings. A particularly relevant example of EMD in high dimensions comes from document retrieval and classification, a topic that received a remarkable amount of attention [KSKW15]. Each document is represented as a collection of vectors given by embedding each of its words into a geometric space [MSC+13, PSM14], and the distance between documents (as their similarity measure) is high-dimensional EMD, aptly named the *Word Mover's Distance*.

1.1 Our Results

We study a natural "divide-and-conquer" algorithm for estimating $\mathrm{EMD}_X(A, B)$, when the underlying metric space is the high-dimensional hypercube $\{0,1\}^d$ with ℓ_1 distance. The algorithm and analysis will be especially clean in this setting, since random coordinate sampling gives a simple partitioning scheme for the hypercube. Our analysis extends, via metric embeddings, to approximating $\mathrm{EMD}_X(A, B)$ when $X = (\mathbb{R}^d, \|\cdot\|_p)$ is an ℓ_p space with $p \in [1, 2]$ (see Appendix E; we focus on $\{0,1\}^d$ with ℓ_1 distance in the rest of the paper due to its simplicity).[1]

At a high level, the algorithm recursively applies randomized space partitions and builds a rooted (randomized) tree. Each point of A and B then goes down to a leaf of the tree, and the matching is formed with a bottom-up approach. The description of the algorithm, named ComputeEMD, is presented in Figure 1. More formally, ComputeEMD(A, B, h) takes three inputs, where h specifies the depth of the tree left and $A, B \subseteq \{0,1\}^d$ but their sizes do not necessarily match. It returns a triple (M, A', B') where M is a *maximal* partial matching between A and B, and $A' \subset A$ and $B' \subset B$ are the unmatched points. (Note that A' and B' will be empty whenever $|A| = |B|$ since M is maximal, and this will be the case in the initial call.) The algorithm samples a random coordinate $\boldsymbol{i} \sim [d]$ in order to partition $\{0,1\}^d$ into two sets (according to the value of coordinate $\boldsymbol{i}$), recursively solves the problem in each set, and matches unmatched points arbitrarily.

Our main result is an improved analysis of the approximation the algorithm achieves.

Theorem 1. *Let $A, B \subset \{0,1\}^d$ be any two multi-sets of size s. Then, with probability at least* 0.9*,*

[1] In particular, [KSKW15] considers high-dimensional EMD over $\mathbb{R}^d$ with ℓ_2. Our results also extend to computing $\mathrm{EMD}_X(\mu_A, \mu_B)$, where μ_A and μ_B are arbitrary distributions supported on size-s sets A and B, respectively, and the goal is to approximate the minimum expectation of $\|\boldsymbol{a} - \boldsymbol{b}\|_1$, where $(\boldsymbol{a}, \boldsymbol{b}) \sim \mu_{A,B}$ for a coupling of μ_A and μ_B.

ComputeEMD (A, B, h)

Input: Two multi-sets $A, B \subseteq \{0,1\}^d$ and a nonnegative integer h.
Output: A tuple (M, A', B'), where M is a maximal matching between A and B, and $A' \subset A$ and $B' \subset B$ are points not participating in the matching M.

1. **Base Case**: If $|A| = 0$ or $|B| = 0$ or $h = 0$, or there is a single point $x \in \{0,1\}^d$ where $A = \{x, \ldots, x\}$ and $B = \{x, \ldots, x\}$, return (M, A', B'), where M is an arbitrary maximal partial matching between A and B, and let $A' \subset A$ and $B' \subset B$ be the left-over points. When $h = 0$, M should match as many identical pairs of points of A and B as possible.

2. **Divide**: Otherwise, sample $\boldsymbol{i} \sim [d]$ uniformly at random, and consider the four subsets

$$A_1 = \{a \in A : a_{\boldsymbol{i}} = 1\} \qquad A_0 = \{a \in A : a_{\boldsymbol{i}} = 0\}$$
$$B_1 = \{b \in B : b_{\boldsymbol{i}} = 1\} \qquad B_0 = \{b \in B : b_{\boldsymbol{i}} = 0\}.$$

   Make two recursively calls: run ComputeEMD$(A_1, B_1, h-1)$ to obtain (M_1, A_1', B_1') and ComputeEMD$(A_0, B_0, h-1)$ to obtain (M_0, A_0', B_0').

3. **Conquer**: Find an arbitrary maximal matching M_L between $A_1' \cup A_0'$ and $B_1' \cup B_0'$, and return $(M_1 \cup M_0 \cup M_L, A', B')$, where A', B' are unmatched points of A, B, respectively.

Figure 1: The ComputeEMD algorithm.

ComputeEMD(A, B, d) *outputs a tuple* $(\mathbf{M}, \emptyset, \emptyset)$*, where* $\mathbf{M}$ *is a matching of* A *and* B *satisfying*

$$\mathrm{EMD}(A, B) \le \sum_{(a,b)\in\mathbf{M}} \|a - b\|_1 \le \tilde{O}(\log s) \cdot \mathrm{EMD}(A, B).$$

Moreover, ComputeEMD *runs in time linear in the input size* sd*.*

The above algorithm is not new, and falls within the framework of hierarchal methods for geometric spaces, building on what is broadly referred to as a *quadtree* [Sam84]. In fact, essentially the same tree structure is studied in [AIK08, BDI+20]. In the rest of the paper we will follow this convention and name the tree a quadtree, in which each node corresponds to a recursive call and is associated with a subcube of $\{0,1\}^d$. Leaves of this tree correspond to the base case in ComputeEMD.

Prior to our work, the analytical framework for studying tree-based algorithms for EMD is the method of randomized tree embeddings [Bar96, Bar98, CCG+98, FRT04]. For example, [AIK08] consider sampling a random quadtree and assigning positive weights to its edges to define a tree metric. By studying the worst-case distortion of the original points in the randomized tree metric (given by the quadtree and weights), the analysis shows that the cost of the bottom-up matching achieves an $O(\min\{\log s, \log d\} \log s)$-approximation to EMD.[2] We note that this method of ran-

[2] In particular, [AIK08] proves a $O(\log d \log s)$-distortion bound for embedding into ℓ_1, which would naively result in a $O(\log d \log s)$-approximation to EMD. We note that even though the *expected* cost of the matching output by the bottom-up approach is an $O(\log d \log s)$-approximation to EMD(A, B), the matching output is an $O(\min\{\log d, \log s\} \log s)$-approximation with probability 0.9. See Appendix A.

domized tree embeddings is the de-facto technique for analyzing tree-based algorithms for EMD in both low- and high-dimensions [Cha02, IT03, Ind07, AIK08, ABIW09, BI14, BDI+20]. It has the added benefit of producing an embedding of EMD into ℓ_1, which immediately implies sketching and nearest neighbor search algorithms.

Our work directly improves on [BDI+20] (their Theorem 3.5 is analogous to Theorem 1 with a bound of $O(\log^2 s)$), who (essentially) study ComputeEMD in the context of similarity search. Especially compelling are their experimental results, which show that the quality of the matching output by the quadtree *significantly* outperforms the cost of that same matching when measured with respect to distances in the tree embedding. This empirical observation is immensely useful in practice, since computing the cost of the matching incurs a relatively minor overhead. From a theoretical perspective, the observation suggests an avenue for improved analyzes, which in particular, should not heavily rely on the worst-case distortion of randomized tree embeddings. Theorem 1 gives a sound theoretical explanation to this empirical observation.

Theorem 1 is tight for ComputeEMD up to $\text{poly}(\log\log s)$ factors (see Appendix B). Conceptually, our analysis goes beyond worst-case distortion of randomized tree embeddings and implies the best linear time approximation for EMD over the hypercube. We note that the proof will not give an embedding into ℓ_1. Rather, Theorem 1 suggests an approach to improved approximations by linear sketches; implementing this approach, however, will require additional work, which we discuss next. Besides algorithms based on the quadtree, another class of algorithms proceed by sampling a geometric spanner [HPIS13] and utilizing min-cost flow solvers [She17]; these algorithms may achieve $O(1/\epsilon)$-approximations but require $ds^{1+\epsilon} \cdot \text{polylog}(s, d)$ time. It is also not clear whether these algorithms and their analysis imply efficient sketching algorithms.

Improved Sketching for EMD. We recall the notion of linear sketches, so that we may discuss one- and two-round sketches. Given an input vector $f \in \mathbb{R}^n$ and a function $\varphi(f) : \mathbb{R}^n \to \mathbb{R}$ that we would like to approximate, a (one-round) linear sketch generates a random matrix $\mathbf{S}_1 \in \mathbb{R}^{k\times n}$ and outputs an approximation $\tilde{\varphi} = \tilde{\varphi}(\mathbf{S}_1, \mathbf{S}_1 f)$ to $\varphi(f)$ based only on the matrix $\mathbf{S}_1$ and vector $\mathbf{S}_1 f$. A two-round linear sketch is allowed to generate a second random matrix $\mathbf{S}_2$, from a distribution depending on $(\mathbf{S}_1, \mathbf{S}_1 f)$, and output $\tilde{\varphi} = \tilde{\varphi}(\mathbf{S}_1, \mathbf{S}_1 f, \mathbf{S}_2, \mathbf{S}_2 f)$. The space of a linear sketch (resp. two-round linear sketch) is the number of bits needed to store $\mathbf{S}_1 f$ (resp. $\mathbf{S}_1 f$ and $\mathbf{S}_2 f$).[3]

For sketching EMD over $\{0,1\}^d$, we encode two size-s multi-sets $A, B \subset \{0,1\}^d$ as an input vector $f = f_{A,B} \in \mathbb{R}^{2\cdot 2^d}$, where for $x \in \{0,1\}^d$, f_x and f_{x+2^d} contains the number of occurrences of $x \in A$ and $x \in B$, respectively. We let $\varphi(f_{A,B}) = \text{EMD}(A, B)$. One- and two-round linear sketches immediately result in one- and two-pass *streaming* algorithms in the *turnstile* model (where insertions and deletions of points are allowed), as well as one- and two-round two-party communication protocols, and one- and two-round MPC algorithms. (See Section 5 and Appendix F for more detail.)

Our first result for sketching EMD shows that it is possible to losslessly capture the approximation of Theorem 1 with a *two-round* linear sketch.

Theorem 2. *There is a two-round linear sketch using* $\text{poly}(\log s, \log d)$ *bits of space which, given a*

[3]For now, consider the random oracle model, where the algorithm is not charged for the space required to store the random matrices $\mathbf{S}_1, \mathbf{S}_2$. Using standard techniques and an application of Nisan's pseudo-random generator, we show that (in our application) this random oracle assumption can be dropped with a small additive $\tilde{O}(d)$ in the space complexity (see Corollaries F.1 and 6.11). The $\tilde{O}(d)$ is a minor overhead given that each update of $A \cup B$ in the streaming model requires $d + 1$ bits to store.

pair of size-s multi-sets $A, B \subset \{0,1\}^d$, outputs $\widehat{\boldsymbol{\eta}} \in \mathbb{R}$ satisfying

$$\mathrm{EMD}(A,B) \le \widehat{\boldsymbol{\eta}} \le \tilde{O}(\log s) \cdot \mathrm{EMD}(A,B) \tag{1}$$

with probability at least $2/3$.

Theorem 2 implies a two-pass streaming algorithm as well as a two-round communication protocol for approximating EMD with the same space and approximation factors. To the best of our knowledge, the prior best sublinear space linear sketch (or streaming algorithm) for *any* number of rounds (or passes) utilizes the ℓ_1-embedding [AIK08] and achieves approximation $O(\min\{\log s, \log d\} \log s)$.

Next we show that the two-round linear sketch of Theorem 2 can be further compressed into a single round, albeit at the cost of a small additive error.

Theorem 3. *Given $\epsilon \in (0,1)$, there is a (one-round) linear sketch using $O(1/\epsilon) \cdot \mathrm{poly}(\log s, \log d)$ bits of space which, given a pair of size-s multi-sets $A, B \subset \{0,1\}^d$, outputs $\widehat{\boldsymbol{\eta}} \in \mathbb{R}$ satisfying*

$$\mathrm{EMD}(A,B) \le \widehat{\boldsymbol{\eta}} \le \tilde{O}(\log s) \cdot \mathrm{EMD}(A,B) + \epsilon s d$$

with probability at least $2/3$.

Notice that $\mathrm{EMD}(A,B) \ge s$ when $A \cap B = \emptyset$, in which case Theorem 3 yields an $\tilde{O}(\log s)$ approximation in $\tilde{O}(d)$ space. More generally, if the *Jaccard Index* of A and B is bounded away from 1, we have the following corollary.

Corollary 1.1. *Given $\epsilon \in (0,1)$ there is a (one-round) linear sketch using $O(d/\epsilon) \cdot \mathrm{poly}(\log s, \log d)$ space which, given size-s $A, B \subset \{0,1\}^d$ such that $|A \cap B|/|A \cup B| \le 1 - \epsilon$, outputs $\widehat{\boldsymbol{\eta}} \in \mathbb{R}$ satisfying*

$$\mathrm{EMD}(A,B) \le \widehat{\boldsymbol{\eta}} \le \tilde{O}(\log s) \cdot \mathrm{EMD}(A,B)$$

with probability at least $2/3$.

Theorem 3 implies a one-pass streaming and a one-round communication protocol using the same space and achieving the same approximation. The proof of Theorem 3 involves the design of several new sketching primitives, which may be of broader interest. Specifically, our linear sketch needs to address the problem of *sampling with meta-data* (see Section 1.2), which can be used to approximate data-dependent weighted ℓ_1-distances. Its analysis extended and generalized error analysis of the precision sampling framework [AKO10, JST11, JW18] to multivariate sampling, and provides new insights into the power of this framework.

1.2 Technical Overview

1.2.1 Analysis of Quadtrees

We start with an overview of the proof of Theorem 1. The running time analysis is straightforward so we focus on (1). The first inequality of (1) holds trivially. It suffices to show that the matching $\mathbf{M}$ returned by ComputeEMD has cost in the original metric upperbounded by $\tilde{O}(\log s) \cdot \mathrm{EMD}(A,B)$ with probability at least 0.9.

Compressed Quadtree and Worst-Case Analysis. The execution of ComputeEMD induces a complete binary tree T_0 of depth d which we refer to as a *Quadtree*[4]. Each internal node is labelled by a coordinate $\boldsymbol{i}$ sampled from $[d]$; its two children correspond to further subdividing $\{0,1\}^d$ in half by fixing coordinate $\boldsymbol{i}$ to be either 0 or 1. We use $S_u \subset \{0,1\}^d$ for each node u to denote the subcube associated with it. After sampling a Quadtree $\mathbf{T}_0$, each point of A and B is assigned to the leaf that contains it and the matching $\mathbf{M}$ is obtained in the bottom-up fashion as in Figure 1.

The first step is a simple compression of the Quadtree $\mathbf{T}_0$. To this end, we only keep the root of $\mathbf{T}_0$ (at depth 0) and its nodes at depths that are powers of 2; we also keep subcubes S_u associated with them. All other nodes are deleted and their adjacent edges are contracted (see Figure 2). The resulting "compressed Quadtree" $\mathbf{T}$ has depth $h = O(\log d)$, where each node u at depth i has $2^{2^{i-1}}$ children and is associated with a subcube $S_u \subseteq \{0,1\}^d$. An execution of ComputeEMD can now be viewed as first drawing a random compressed Quadtree $\mathbf{T}$, assigning points to its leaves, and then building $\mathbf{M}$ from bottom up.

Compressed Quadtrees are essentially the same as tree structures used in [AIK08, BDI+20]. It will be helpful to begin with a recap of the analysis of tree embedding of [AIK08] before presenting an overview of our new techniques.[5] [AIK08] assigns a weight of $d/2^i$ to each edge from a node at depth i to a node at depth $i+1$ in a compressed Quadtree $\mathbf{T}$. This defines a metric embedding $\varphi : A \cup B \to \mathbf{T}$ by mapping each point to a leaf of $\mathbf{T}$. Their analysis bounds (from both sides) the cost of the bottom-up matching in the resulting tree metric. Their choice of edge weights is motivated by the observation that two points $x, y \in \{0,1\}^d$ with $\|x-y\|_1 = d/2^i$ are expected to have their paths diverge for the first time at depth i. If this is indeed the case then $d_{\mathbf{T}}(\varphi(x), \varphi(y))$ would be $\|x-y\|_1$ up to a constant.

To analyze the cost of the bottom-up matching in $\mathbf{T}$, one studies the distortion of this embedding. Firstly, for any $\lambda > 1$ and $x, y \in \{0,1\}^d$, it is easy to verify that

$$\Pr_{\mathbf{T}}\left[d_{\mathbf{T}}(\varphi(x),\varphi(y)) < \frac{1}{\lambda}\cdot\|x-y\|_1\right] \le \left(1-\frac{\|x-y\|_1}{d}\right)^{1+2+\cdots+2^{\left\lfloor \log_2\left(\frac{\lambda d}{\|x-y\|_1}\right)\right\rfloor}} \le 2^{-\Omega(\lambda)}.$$

Thus by a union bound, for all $x, y \in A \cup B$ we have

$$d_{\mathbf{T}}(\varphi(x),\varphi(y)) \ge \Omega\left(\frac{1}{\log s}\right)\cdot\|x-y\|_1 \tag{2}$$

with probability at least $1 - 1/\mathrm{poly}(s)$, which essentially means that we can assume (2) in the worst case. Thus, the cost of the bottom-up matching in the tree metric is at least $\Omega(1/\log s)\cdot\mathrm{EMD}(A,B)$. Furthermore, given points x, y with $\|x-y\|_1 = \Theta(d/2^j)$, the probability that paths of x, y diverge at level $j-k$ is at $\Theta(2^{-k})$ for each k, and when it does, $d_{\mathbf{T}}(\varphi(x),\varphi(y)) = \|x-y\|_1\cdot\Theta(2^k)$. Since $j \le h = O(\log d)$,

$$\mathbf{E}\left[d_{\mathbf{T}}(\varphi(x),\varphi(y))\right] \le \|x-y\|_1 + \sum_{k=0}^{j}\Theta(2^{-k})\cdot\|x-y\|_1\cdot\Theta(2^k) = O(\log d)\cdot\|x-y\|_1. \tag{3}$$

[4]Note that the definition given here is slightly different from its formal definition at the beginning of Section 3, where a Quadtree will have depth $2d$. This difference is not important for the overview.

[5]We refer the reader to a formal treatment of ComputeEMD via tree embeddings in Appendix A.

This, combined with the fact that the bottom-up matching is *optimal* for a tree metric, implies that the expected cost of the bottom-up matching in the tree metric is at most $O(\log d) \cdot \mathrm{EMD}(A, B)$.[6] Together they yield the aforementioned $O(\log s \log d)$ approximation [AIK08]. In fact, the analysis is tight: see Appendix C for instances where the cost of the matching in the tree metric can be both a $(\log d)$-factor larger and a $(1/\log s)$-factor smaller than the EMD.

The Challenge of Worst-Case Distortion. Let $A_v = \{a \in A \colon a \in S_v\}$ be the set of points in A that pass through v, and let B_v be defined similarly. Then (2) upperbounds the diameter of $A_v \cup B_v$ in the original metric with high probability, i.e., every $x, y \in A_v \cup B_v$ has their distance $\|x - y\|_1$ bounded by $O(\log s) \cdot (d/2^i)$ with high probability if v is at depth i. Intuitively, upper bounds on diameters of $A_v \cup B_v$ can be very helpful for bounding the cost of the bottom-up matching in the original metric: If $a \in A$ and $b \in B$ are matched by ComputeEMD and their root-to-leaf paths diverge first at v, then we can use the diameter of v to bound their contribution $\|a - b\|_1$. However, as suggested by instances in Appendix C, the loss of $O(\log s)$ in (2) is the best one can hope for in the worst-case.

Since the approximation we aim for is $\tilde{O}(\log s)$, it seems considering worst-case diameter may be sufficient; however, (3) naively results in an additional $O(\log d)$-factor loss in the approximation. Suppose that $a \in A$ and $b \in B$ are matched by the optimal matching and are at distance $\|a - b\|_1 = \Theta(d/2^j)$. As in (3), for every $k \le j$ the root-to-leaf paths of a and b may diverge at depth $j - k$ with probability $\Theta(2^{-k})$. If, once a and b diverge at depth $j - k$, a is matched to a point within distance $d/2^{j-k} \cdot O(\log s)$ (using the worst-case bound on the radii at depth $j - k$), then once we evaluate the expectation of $\|a - \mathbf{M}(a)\|_1$, where $\mathbf{M}(a) \in B$ is the matching produced by ComputeEMD, we would get $O(\log d \log s) \cdot d/2^j = O(\log d \log s) \cdot \|a - b\|_1$.

Inspector Payments. We setup the analytical framework to go beyond worst-case distortion, and give the main geometric insights next. Our analytical strategy is similar, in spirit, to the "accounting method" in amortized analysis. We consider a protocol carried out by an *Inspector*, who knows A,B, as well as the optimal matching M^*, and the *Algorithm*, who executes ComputeEMD(A, B, d), building the compressed Quadtree $\mathbf{T}$ and the bottom-up matching $\mathbf{M}$. As the algorithm executes, the inspector monitors the execution, and makes payments to the algorithm using knowledge of M^*. Specifically, the inspector will track each pair $(a, b) \in M^*$, and pays $\mathbf{Pay}_{\mathbf{T}}(a)$ for a as well as a payment $\mathbf{Pay}_{\mathbf{T}}(b)$ for b. The protocol effectively produces a coupling between the cost of the bottom-up matching $\mathbf{M}$ from $\mathbf{T}$, and the payments the inspector makes; we show that total payment can always cover the cost of $\mathbf{M}$, and is at most $\tilde{O}(\log s) \cdot \mathrm{EMD}(A, B)$ with probability 0.9.

Formally we let $(v_0(x), v_1(x), \ldots, v_h(x))$ denote the root-to-leaf path of x in T. For each $(a, b) \in M^*$,

$$\mathbf{Pay}_T(a) \stackrel{\text{def}}{=} \sum_{i \in [h]} \mathbf{1}\{v_i(a) \neq v_i(b)\} \Big(\|a - c_{v_i(a)}\|_1 + \|a - c_{v_{i-1}(a)}\|_1 \Big), \tag{4}$$

where c_v denotes the center-of-mass of $A_v \cup B_v$:

$$c_v \stackrel{\text{def}}{=} \frac{1}{|A_v \cup B_v|} \sum_{x \in A_v \cup B_v} x,$$

and $\mathbf{Pay}_T(b)$ is defined similarly. Intuitively, this payment strategy corresponds to an inspector who tracks each pair $(a, b) \in M^*$, and whenever a and b first diverge in the tree at node v, pays

[6] The reason that this analysis can achieve approximation $O(\min\{\log s, \log d\} \log s)$, as opposed to $O(\log d \log s)$ is that with probability $1 - 1/s$, every $x, y \in A \cup B$ with $\|x - y\|_1 = \Theta(d/2^j)$ diverges at depth after $j - O(\log s)$.

twice the distance between a (as well as b) and to the center-of-mass along the v-to-leaf paths of a (as well as b).

In Lemmas 3.3 and 3.4, we show that for any T, the total inspector payments is sufficient to cover the cost of the matching $\mathbf{M}$ produced by the Quadtree:

$$\mathbf{Cost}(\mathbf{M}) = \sum_{(a,b)\in\mathbf{M}} \|a-b\|_1 \le \sum_{(a,b)\in M^*} \mathbf{Pay}_T(a) + \mathbf{Pay}_T(b). \tag{5}$$

The inspector payments (4) depend on the data A, B, as well as a compressed tree T in two ways. The first is the depth when $(a,b) \in M^*$ first diverge, captured by the indicator $\mathbf{1}\{v_i(a) \neq v_i(b)\}$. The second is the distance between a to centers-of-mass, which not only depends on (a,b), but also on global properties of $A \cup B$. At a high level, incorporating this second aspect is the main novelty, since the distance between a and the center-of-mass of $A_v \cup B_v$ is an *average* notion of radii at v. In particular, the distance between a and the center-of-mass at v is at most the average distance between a and points in $A_v \cup B_v$.[7] Therefore, if the inspector pays a large amount, then an average point in $A_v \cup B_v$ is far from a (as opposed to the farthest point implied by worst-case radii).

Bounding Inspector Payments. The technically most challenging part is upperbounding the expected total inspector payment (5), over a random compressed $\mathbf{T}$, by $\tilde{O}(\log s) \cdot \mathrm{EMD}(A,B)$. For the remainder of this subsection, consider a fixed $(a,b) \in M^*$ at distance $\|a-b\|_1 = \Theta(d/2^j)$, and we will upperbound the expectation of $\mathbf{Pay}_{\mathbf{T}}(a) + \mathbf{Pay}_{\mathbf{T}}(b)$ for random $\mathbf{T}$; furthermore, consider the payment in $\mathbf{Pay}_{\mathbf{T}}(a)$ incurred from depth i after a and b have diverged. Namely, by linearity of expectation, we want to upper bound $\mathbf{E}[\|a - \mathbf{c}_{\mathbf{v}_i(a)}\|_1]$ for each depth i, where the expectation is over $\mathbf{T}$ conditioned on already having diverged from b (the conditioning will not heavily influence the geometric intuition, so we will forget about this for the rest of this overview). Let $\mathbf{v}_i = \mathbf{v}_i(a)$ be the vertex at depth i containing a, and let $\mathbf{c}_i = \mathbf{c}_{\mathbf{v}_i}$.

Similar to the worst-case bounds on radii, $\mathbf{E}[\|a - \mathbf{c}_i\|_1]$ may still be $d/2^i \cdot \Omega(\log s)$. For instance, let i_1 be a relatively large depth and for small $\epsilon \approx 10^{-6}$, consider a set P_1 of s^ϵ points at distance $\epsilon \log s \cdot d/2^{i_1}$ around a. Then, at depth i_1 of a random tree $\mathbf{T}$, a point in P_1 traverses down to node $\mathbf{v}_{i_1}$ with non-negligible probability, roughly $1/s^{-\epsilon}$. If no other points lie closer to a than P_1, then $\mathbf{E}[\|a - \mathbf{c}_{i_1}\|_1] = d/2^{i_1} \cdot \Omega(\epsilon \log s)$, since in expectation, some points from P_1 make it to $\mathbf{v}_{i_1}$ and move the center-of-mass $\mathbf{c}_{i_1}$ away from a. If this happened for every depth i, the inspector would be in trouble, as there are $O(\log d)$ levels and a similar argument to that of worst-case radii would mean they would pay $O(\log d \log s) \cdot \|a-b\|_1$ in expectation.

However, we claim that if the arrangement of P_1 resulted in $\mathbf{E}[\|a - \mathbf{c}_{i_2}\|_1] = d/2^{i_1} \cdot \Omega(\epsilon \log s)$, the same situation will be more difficult to orchestrate for depth $i_2 \le i_1 - O(\log\log s)$. In particular, at depth i_2, in order to have $\mathbf{E}[\|a - \mathbf{c}_{i_2}\|_1] = d/2^{i_2} \cdot \Omega(\epsilon \log s)$, there must be a set of points P_2 at distance $d/2^{i_2} \cdot \Omega(\epsilon \log s)$ which will cause $\mathbf{c}_{i_2}$ to be far from a. However, it is no longer enough to have $|P_2| = s^\epsilon$. The reason is that points of P_1 in $\mathbf{v}_{i_2}$ move the center-of-mass towards a. Since points in P_1 are at distance $\epsilon \log s \cdot d/2^{i_1} \ll d/2^{i_2}$ from a, there will oftentimes be $\Omega(s^\epsilon)$ points from P_1 in $\mathbf{v}_{i_2}$. In order to significantly affect the center-of-mass $\mathbf{c}_{i_2}$, $\mathbf{v}_{i_2}$ must oftentimes have at least $s^\epsilon/\mathrm{polylog}(s)$ points from P_2; otherwise, $\mathbf{c}_{i_2}$ will be mostly an average of points in P_1. Since any given point from P_2 traverses down to $\mathbf{v}_{i_2}$ with probability roughly $1/s^\epsilon$, we must have $|P_2| \ge s^{2\epsilon}/\mathrm{polylog}(s)$. This

[7] This used the fact that ℓ_1 is a normed space.

argument can only proceed for at most $O(1/\epsilon)$ depths before $|P_{O(1/\epsilon)}| > 2s$, in which case we obtain a contradiction, since the points $P_1, \ldots, P_{O(1/\epsilon)}$ must be in $A \cup B$.

Generally, in order to move the center-of-mass $\mathbf{c}_i$ of a vertex $\mathbf{v}_i$ away from a *multiple times* as the depth i goes down, the number of points around a at increasing distances must grow very rapidly. More specifically, we show that if a depth i is "bad," meaning that $\mathbf{E}[\|a - \mathbf{c}_i\|_1] \geq \alpha \cdot d/2^i$ for some $\alpha = \omega(\log\log s)$, then the number of points within a ball of radius $d/(2^i \log s)$ around a and within a larger ball of radius $O(\log s \cdot d/2^i)$ around a must have increased by a factor of $\exp(\Omega(\alpha))$; this means the number of such depths i is at most $((\log s)/\alpha) \cdot \text{poly}(\log\log s)$. Combining this analysis and the fact that a and b must diverge in order to incur payment from the inspector, we obtain our upper bound $\mathbf{E}[\mathbf{Pay}_{\mathbf{T}}(a) + \mathbf{Pay}_{\mathbf{T}}(b)] = \tilde{O}(\log s) \cdot \|a - b\|_1$.

1.2.2 Sketching Algorithms

At a high level, our approach to sketching will involve sampling a compressed Quadtree $\mathbf{T}$ and sketching approximations to the inspector payments. As demonstrated by Theorem 1, these provide a good approximation to $\text{EMD}(A, B)$. The first step is to consider a modified inspector strategy $\boldsymbol{\mathcal{I}}$ which is *oblivious* to M^*, while still achieving

$$\text{EMD}(A, B) \leq \boldsymbol{\mathcal{I}} \leq \tilde{O}(\log s) \cdot \text{EMD}(A, B) \tag{6}$$

with probability at least 0.9 over $\mathbf{T}$, and approximate $\boldsymbol{\mathcal{I}}$ instead. (This $\boldsymbol{\mathcal{I}}$ will be what we call the cost of a compressed Quadtree in Section 3.) More specifically, let E_i denote the set of directed edges from depth $i-1$ to i in a compressed Quadtree $\mathbf{T}$, and let $\boldsymbol{\mathcal{I}} = \sum_{i \in [h]} \boldsymbol{\mathcal{I}}_i$ where

$$\boldsymbol{\mathcal{I}}_i \stackrel{\text{def}}{=} \sum_{(u,v) \in E_i} \big||A_v| - |B_v|\big| \cdot \|c_v - c_u\|_1. \tag{7}$$

The upper bound in (6) follows from upper bounding $\boldsymbol{\mathcal{I}}$ by $\sum_{(a,b) \in M^*} (\mathbf{Pay}_{\mathbf{T}}(a) + \mathbf{Pay}_{\mathbf{T}}(b))$,[8] and the lower bound follows from Lemma 3.3. Intuitively $\boldsymbol{\mathcal{I}}$ measures the cost of moving all unmatched points in $A_v \cup B_v$ from the center-of-mass c_v up to the center-of-mass c_u of the parent of v. A consequence of the proof of Theorem 1 is that we can truncate $\|c_v - c_u\|_1$ and consider

$$w_{u,v} \stackrel{\text{def}}{=} \begin{cases} \frac{d \log s}{2^i} & \|c_v - c_u\|_1 \gg \frac{d \log s}{2^i} \\ \frac{d}{2^i} & \|c_v - c_u\|_1 \ll \frac{d}{2^i} \\ \|c_v - c_u\|_1 & \text{o.w.} \end{cases}, \tag{8}$$

since replacing $\|c_v - c_u\|_1$ with $w_{u,v}$ in (7) will still satisfies (6) with probability at least 0.89. Every edge (u, v) satisfies $\|c_v - c_u\| = O(\frac{d \log s}{2^i})$ with high probability over a random compressed Quadtree $\mathbf{T}$, and considering $d/2^i$ whenever $\|c_v - c_u\|_1 \ll d/2^i$ only adds an additive $O(\log s) \cdot \text{EMD}(A, B)$ to $\boldsymbol{\mathcal{I}}$ (both are consequences of a simple analysis using worst-case distortion).

Both our one and two round sketching algorithms proceed as follows. First, consider drawing a random compressed Quadtree $\mathbf{T}$, and then for each $i \in \{0, \ldots, h-1\}$ define the (implicit) vector

[8] This can be seen by two applications of the triangle inequality, one to upper bound $||A_v| - |B_v||$ by $\sum_{(a,b) \in M^*} \mathbf{1}\{v_i(a) \neq v_i(b) \text{ and } v_i(a) = v \text{ or } v_i(b) = v\}$, and the second by $\|c_{v_i(a)} - c_{v_{i-1}(a)}\|_1 \leq \|a - c_{v_i(a)}\|_1 + \|a - c_{v_{i-1}(a)}\|_1$.

$\Delta^i \in \mathbb{R}^{E_i}$, indexed by directed edges from depth $i-1$ to depth i, given by

$$\Delta^i_{u,v} \stackrel{\text{def}}{=} |A_v| - |B_v|.$$

Now consider the distribution $\mathcal{D}_i$, supported on edges E_i, given by sampling $(u,v) \in E_i$ with probability $|\Delta^i_{u,v}|/\|\Delta^i\|_1$. Then for every $i \in \{0,\dots,h-1\}$, we have

$$\boldsymbol{\mathcal{I}}_i = \|\Delta^i\|_1 \cdot \mathop{\mathbf{E}}_{(\boldsymbol{u},\mathbf{v})\sim\mathcal{D}_i}[w_{\boldsymbol{u},\mathbf{v}}]. \tag{9}$$

Note that we can estimate $\|\Delta^i\|_1$ with an ℓ_1-sketch [Ind06a], and produce a sample $(\boldsymbol{u},\mathbf{v}) \sim \mathcal{D}_i$ via ℓ_1-sampling sketches [AKO10, JST11, JW18]. Furthermore, once we fix a sample $(u,v) \in E_i$, we may estimate $w_{u,v}$ with ℓ_1-sketches as well. Specifically, the sketch of $\|c_u - c_v\|_1$ proceeds by storing $|A_u|$, $|A_v|$, $|B_u|$, and $|B_v|$ using $O(\log s)$ bits, and storing ℓ_1-linear sketches of

$$\chi_{A,u} \stackrel{\text{def}}{=} \sum_{a\in A_u} a, \quad \chi_{A,v} \stackrel{\text{def}}{=} \sum_{a\in A_v} a, \quad \chi_{B,u} \stackrel{\text{def}}{=} \sum_{b\in B_u} b, \quad \text{and} \quad \chi_{B,v} \stackrel{\text{def}}{=} \sum_{b\in B_v} b. \tag{10}$$

Since $c_u = (\chi_{A,u} + \chi_{B,u})/(|A_u| + |B_u|)$ and $c_v = (\chi_{A,v} + \chi_{B,v})/(|A_v| + |B_v|)$, ℓ_1-linear sketches of the above quantities suffice for estimating $\|c_u - c_v\|_1$.

We notice that by definition (8), $w_{\boldsymbol{u},\mathbf{v}}$ is always bounded within $d/2^i$ and $d\log s/2^i$, so that we may estimate (9) by sampling $\operatorname{polylog}(s)$ times from $\mathcal{D}_i$ and computing the empirical average of $w_{\boldsymbol{u},\mathbf{v}}$. This plan is straight-forward to implement with two-rounds of linear sketching. In the first round, we produce the $t = \operatorname{polylog}(s)$ samples $(\boldsymbol{u}_1,\mathbf{v}_1),\dots,(\boldsymbol{u}_t,\mathbf{v}_t) \sim \mathcal{D}_i$, as well as an estimate of $\|\Delta^i\|_1$. Then in the second round, we produce estimates of $w_{\boldsymbol{u}_1,\mathbf{v}_1},\dots,w_{\boldsymbol{u}_t,\mathbf{v}_t}$ given knowledge of $(\boldsymbol{u}_1,\mathbf{v}_1),\dots,(\boldsymbol{u}_t,\mathbf{v}_t)$ by the linear sketches of the vectors in equation (10).

The remaining challenge, however, is to produce $(\boldsymbol{u},\mathbf{v}) \sim \mathcal{D}_i$ and an estimate of $w_{\boldsymbol{u},\mathbf{v}}$ *simultaneously*. We call this problem *sampling with meta-data*, since the ℓ_1-linear sketches of (10) will be the meta-data of the sample $(\boldsymbol{u},\mathbf{v}) \sim \mathcal{D}_i$ needed to reconstruct $w_{\boldsymbol{u},\mathbf{v}}$.

Sampling with Meta-Data and One-Round Sketching. The key task of sampling with meta-data is the following: for $n,k \in \mathbb{N}$, we are given a vector $x \in \mathbb{R}^n$ and collection of *meta-data* vectors $\lambda_1,\lambda_2,\dots,\lambda_n \in \mathbb{R}^k$, and the goal is to sample $i \in [n]$ with probability $|x_i|/\|x\|_1$ (or more generally, $|x_i|^p/\|x\|_p^p$), and output both i and an approximation $\widehat{\lambda}_i \in \mathbb{R}^k$ of the vector λ_i. The challenge is to solve this problem with a small-space linear sketches of x and the meta-data vectors $\lambda_1,\dots,\lambda_n$. It is not hard to see that sampling with meta-data is exactly the problem we seek to solve for linear sketching of EMD.[9] We refer the reader to Appendix D for generic bounds on sampling from meta-data. However, our application to EMD for Theorem 3 will require significant additional techniques, which we detail next.

Our algorithm builds on a powerful sketching technique known as *precision sampling* [AKO10, JST11, JW18] for sampling an index $i \in [n]$ proportional to $|x_i|/\|x\|_1$ for a vector $x \in \mathbb{R}^n$ (or more generally, for $|x_i|^p/\|x\|_p^p$, but we focus on $p=1$). The idea is to produce, for each $i \in [n]$

[9] Namely, x is the vector Δ^i (after hashing the universe E_i to size $\operatorname{poly}(s)$, since Δ^i is a $2s$-sparse vector), and the meta-data vectors are the ℓ_1-linear sketches of (10) for each $(u,v) \in E_i$. In other words, $n = \operatorname{poly}(s)$, and $k = \operatorname{polylog}(s,d)$.

an independent exponential random variable $\boldsymbol{t}_i \sim \text{Exp}(1)$, and construct a "scaled vector" $\boldsymbol{z} \in \mathbb{R}^n$ with coordinates $\boldsymbol{z}_i = x_i/\boldsymbol{t}_i$. One then attempts to return the index $i_{\max} = \text{argmax}_{i\in[n]} \boldsymbol{z}_i$, since

$$\mathop{\mathbf{Pr}}_{\boldsymbol{t}_1,\ldots,\boldsymbol{t}_n\sim\text{Exp}(1)}\left[\operatorname*{argmax}_{i'\in[n]} \frac{|x_{i'}|}{\boldsymbol{t}_{i'}} = i\right] = \frac{|x_i|}{\|x\|_1}.$$

To find the the index $i_{\max}$ with a linear sketch, we can use a "heavy-hitters" algorithm, such as the Count-Sketch of [CCFC02]. Specifically, Count-Sketch with error $\epsilon \in (0,1)$ allows us to recover an estimate $\tilde{\boldsymbol{z}}$ to $\boldsymbol{z}$ satisfying (roughly) $\|\tilde{\boldsymbol{z}} - \boldsymbol{z}\|_\infty \le \epsilon\|\boldsymbol{z}\|_2$. Then one can show that $\text{argmax}_{i'\in[n]} |\tilde{\boldsymbol{z}}_{i'}|$ is close to being distributed as $|x_i|/\|x\|_1$.[10]

In order to sample with meta-data, our sketch similarly samples independent exponential $\boldsymbol{t}_1, \ldots, \boldsymbol{t}_n \sim \text{Exp}(1)$ and applies a Count-Sketch data structure on $\boldsymbol{z} \in \mathbb{R}^n$, where $\boldsymbol{z}_i = x_i/\boldsymbol{t}_i$, and obtains an estimate $\tilde{\boldsymbol{z}}$ of $\boldsymbol{z}$. In addition, for each $\ell \in [k]$, we apply a Count-Sketch data structure with error ϵ for the vector $\boldsymbol{w}^\ell \in \mathbb{R}^n$ given by the ℓ-th coordinates of the meta-data vectors $\lambda_i/\boldsymbol{t}_i$, namely $\boldsymbol{w}^\ell_i = (\lambda_i)_\ell/\boldsymbol{t}_i$.[11] From this we obtain an estimate $\tilde{\boldsymbol{w}}^\ell$ of $\boldsymbol{w}^\ell$. The insight is the following: suppose the sample produced is $i^* \in [n]$, which means it satisfies $\tilde{\boldsymbol{z}}_{i^*} \approx \max_{i\in[n]} |x_i|/\boldsymbol{t}_i$. Then the value $\boldsymbol{t}_{i^*}$ should be relatively small: in particular, we expect $\boldsymbol{t}_{i^*}$ to be $\Theta(|x_{i^*}|/\|x\|_1)$. When this occurs, for each $\ell \in [k]$, the guarantees of Count-Sketch imply that the estimate $\boldsymbol{t}_{i^*} \cdot \tilde{\boldsymbol{w}}^\ell_{i^*}$ satisfies

$$\left|\boldsymbol{t}_{i^*} \cdot \tilde{\boldsymbol{w}}^\ell_{i^*} - (\lambda_{i^*})_\ell\right| = \boldsymbol{t}_{i^*}\left|\tilde{\boldsymbol{w}}^\ell_{i^*} - \boldsymbol{w}^\ell_{i^*}\right| \le \epsilon \boldsymbol{t}_{i^*}\|\boldsymbol{w}^\ell\|_2 \left(= O\left(\epsilon|x_{i^*}| \cdot \frac{\|(\lambda_\cdot)_\ell\|_1}{\|x\|_1}\right) \text{ in expectation}\right)$$

where $(\lambda_\cdot)_\ell \in \mathbb{R}^n$ is the vector of ℓ-th coordinates of the meta-data $\lambda_1, \ldots, \lambda_n$. In other words, if the size of $(\lambda_{i^*})_\ell$ is comparable to $|x_{i^*}|$, and if the ratio $\|(\lambda_\cdot)_\ell\|_1/\|x\|_1$ of the meta-data norms to the norm of x is bounded, then $\boldsymbol{t}_{i^*}\tilde{\boldsymbol{w}}^\ell_{i^*}$ is a relatively good approximation to $(\lambda_{i^*})_\ell$. Repeating the above argument for every $\ell \in [k]$ recovers an approximation for $\lambda_{i^*} \in \mathbb{R}^k$.

In our application, the vector $x \in \mathbb{R}^n$ is given by Δ^i, and the meta-data are the ℓ_1-linear sketches of (10), as well as counts $|A_v|, |A_u|, |B_v|$, and $|B_u|$ needed to reconstruct $w_{u,v}$. At a high level, applying the above algorithm results in an ℓ_1-sample $(\boldsymbol{u}, \mathbf{v}) \sim \mathcal{D}_i$, and an approximation to the ℓ_1-linear sketches of (10) and approximations to the counts $|A_u|, |A_v|, |B_u|$, and $|B_v|$, where the error guarantee is additive and linearly related to the ratio between the sum of magnitudes of the coordinates we seek to recover and $\|\Delta^i\|_1$ (this plan will runs into a technical issue, which we will soon expand on).

In order to see how this may be implemented, consider the problem of recovering the ℓ_1-linear sketch of $\chi_{A,v}$. The ℓ_1-sketch of [Ind06a] has coordinates given by inner products of (10) with vectors of independent Cauchy random variables, which means that the ℓ-th coordinate in the ℓ_1-linear sketch of $\chi_{A,v}$ is given by

$$\frac{1}{\text{median}(|\mathcal{C}|)} \sum_{j=1}^{d} \mathbf{C}_j \left(\sum_{a\in A_v} a_j\right),$$

[10] We remark that prior works [AKO10, JST11, JW18] using this framework for ℓ_p sampling require an additional statistical test, where the algorithm outputs **FAIL** when the test fails. The statistical test adds a layer of complexity to the presentation of an otherwise elegant algorithm, and we observe in Lemma 6.5 that such a statistical test is unnecessary, up to minor errors in sampling distribution.

[11] Since the Count-Sketch data structures will be of size $\text{polylog}(n)$, the final size of the sketch is $k \cdot \text{polylog}(n)$.

where $\mathbf{C}_1, \ldots, \mathbf{C}_d$ are independent Cauchy random variables $\mathcal{C}$ and median($|\mathcal{C}|$) is the median of the distribution.[12] From the fact that $\mathbf{C}_1, \ldots, \mathbf{C}_d$ are Cauchy random variables, the above sum is expected to have magnitude $O(\log d) \cdot O(d) \cdot |A_v|$. Since the error guarantee depends on the sum of magnitudes across the ℓ-th coordinate of all meta-data vectors (one for each edge in E_i), if we sample $(\boldsymbol{u}, \mathbf{v}) \sim \mathcal{D}_i$, the additive error on the ℓ-th coordinate of the ℓ_1-sketch of $\chi_{A,\mathbf{v}}$ we recover becomes

$$\epsilon \cdot \frac{|\Delta^i_{\boldsymbol{u},\mathbf{v}}|}{\|\Delta^i\|_1} \cdot O(\log d) \cdot O(d) \cdot \sum_v |A_v| = O\left(\epsilon \cdot sd \log d \cdot \frac{|\Delta^i_{\boldsymbol{u},\mathbf{v}}|}{\|\Delta^i\|_1}\right), \tag{11}$$

where we used the fact that $\{A_v\}_v$ partitions A in order to say $\sum_v |A_v| = s$. Furthermore, if $\|\Delta^i\|_1 \ll \epsilon_0 s 2^i/(\log d \log s)$, then from (9) and the fact $w_{u,v} \leq d \log s/2^i$, we can already conclude that $\boldsymbol{\mathcal{I}}_i \ll \epsilon_0 sd/\log d$, which is a negligible (since we allow $\epsilon_0 sd$ additive error, and there are at most $O(\log d)$ depths), so that we may assume $\|\Delta^i\|_1 = \tilde{\Omega}(\epsilon s 2^i)$. For these depths where $\|\Delta^i\|_1$ is sufficiently large, the additive error in (11) incurred will be smaller than a typical ℓ-th coordinate of the sketch of $\chi_{A,\mathbf{v}}$, giving us a coordinate-wise relative error of the sketch.

The above is sufficient for recovering a relative error approximation to the ℓ_1-linear sketch of $\chi_{A,v}$ and $\chi_{B,v}$ for the vertices v at depth i, by setting ϵ to be a small enough $1/\text{polylog}(s, d)$. However, the same argument loses additional $O(s)$-factors when recovering approximations to ℓ_1-linear sketches of $\chi_{A,u}$ and $\chi_{B,u}$ for the vertices u at depth $i-1$. The reason is that the size of $\|(\lambda.)_\ell\|_1$ becomes $O(\log d) \cdot O(d) \cdot \sum_v |A_{\pi(v)}|$, where $\pi(v)$ is the parent of v; in some cases, this is $\Omega(s^2 d \log d)$. Intuitively, the problem is that while the the multi-sets $\{A_v\}_v$ and $\{B_v\}_v$ partition A and B, the multi-sets $\{A_{\pi(v)}\}_v$ and $\{B_{\pi(v)}\}_v$ may duplicate points s times, causing the magnitudes of coordinates in the ℓ_1-linear sketches to much larger. This additional factor of $O(s)$ would require us to make $\epsilon = 1/(s \cdot \text{polylog}(s, d))$, increasing the space complexity of the sketch to $\tilde{O}(s)$, effectively rendering it trivial.[13]

To get around this issue, we utilize a two-step precision sampling with meta-data algorithm. We first sample $\boldsymbol{u}^*$ with probability proportional to the ℓ_1-norm of Δ^i restricted to coordinates corresponding to the children of $\boldsymbol{u}^*$; namely, we sample $\boldsymbol{u}^*$ with probability proportional to $\sum_{v:\pi(v)=\boldsymbol{u}^*} |\Delta^i_{\boldsymbol{u}^*,v}|$.[14] Since the multi-sets $\{A_u\}_u$ and $\{B_u\}_u$ partition A and B, and we can recover ℓ_1-linear sketches for $\chi_{A,\boldsymbol{u}^*}$ and $\chi_{B,\boldsymbol{u}^*}$ up to relative error, as well as approximations to the counts for $|A_{\boldsymbol{u}^*}|$ and $|B_{\boldsymbol{u}^*}|$. Once we have $\boldsymbol{u}^*$, we apply precision sampling with meta-data once more, to sample a child $\mathbf{v}^*$ from $\boldsymbol{u}^*$ proportional to $|\Delta^i_{\boldsymbol{u}^*,\mathbf{v}^*}|$. Specifically, we generate a second set of exponentials $\{\boldsymbol{t}_v\}_v$, one for each child node v. In order to ensure that the sample produced by the second sketch actually returns a child $\mathbf{v}^*$ of $\boldsymbol{u}^*$, and not a child of some other node, we crucially must scale the vector Δ^i by *both* the child exponentials $\{\boldsymbol{t}_v\}_v$ as well as the parent exponentials $\{\boldsymbol{t}_u\}_u$ from the first sketch. Thus, we analyze the twice-scale vector $\boldsymbol{z}$ with coordinates $\boldsymbol{z}_v = \Delta^i_{u,v}/(\boldsymbol{t}_u \boldsymbol{t}_v)$, and attempt to find the largest coordinate of $\boldsymbol{z}$. Importantly, notice that this makes the scaling factors in $\boldsymbol{z}_v$ no longer independent: two children of the same parent share one of their scaling factors. Moreover, the Cauchy random variables used in the ℓ_1-linear sketches must also be the same for children with the same parent. Executing this plan requires a careful analysis of the behavior of norms of

[12] Namely, median($|\mathcal{C}|$) = $\sup\{t \in \mathbb{R} : \mathbf{Pr}_{\mathbf{C}\sim\mathcal{C}}[|\mathbf{C}| \leq t] \leq 1/2$.

[13] In particular, a $\tilde{O}(s/\epsilon^2)$-size sketch may proceed by taking ℓ_1-sketches of the s vectors in A and B such that all pairwise distances are preserved, giving a $(1+\epsilon)$-approximation to EMD(A, B).

[14] This value must be approximated via a Cauchy sketch, since the ℓ_1 norm of the children is norm of $\boldsymbol{u}^*$ is not a linear function.

vectors scaled by several non-independent variables, as well as a nested Count-Sketch to support the two-stage sampling procedure.

2 Preliminaries

Given $n \geq 1$ we write $[n]$ to denote $\{1, \ldots, n\}$. Given a vector $x \in \mathbb{R}^n$ and a real $t \geq 0$, we define $x_{-t} \in \mathbb{R}^n$ to be the vector obtained by setting the largest $\lfloor t \rfloor$ coordinates of x in magnitude equal to 0. For $a, b \in \mathbb{R}$ and $\epsilon \in (0, 1)$, we use the notation $a = (1 \pm \epsilon)b$ to denote the containment of $a \in [(1-\epsilon)b, (1+\epsilon)b]$.

Fix $s, d \in \mathbb{N}$, and consider two multi-sets $A, B \subset \{0,1\}^d$ of size $|A| = |B| = s$. We write

$$\mathbf{Cost}(M) = \sum_{(a,b) \in M} \|a - b\|_1$$

to denote the cost of a matching M between A and B (in the original ℓ_1 distance). For convenience, we will assume without loss of generality that d is a power of 2 and write $h = \log 2d = \log d + 1$. Given a vertex v in a rooted tree T, if v is not the root then we use the notation $\pi(v)$ to denote the parent of v in T.

2.1 Quadtrees and Compressed Quadtrees

We start with a formal definition of Quadtrees used in this paper:

Definition 2.1 (Quadtrees)**.** *A Quadtree is a complete binary tree T_0 of depth $2d$. We say a node v is at depth j if there are $j+1$ nodes on the root-to-v path in T_0 (so the root is at depth 0 and its leaves are at depth $2d$). Each internal node of T_0 is labelled a coordinate $i \in [d]$, and we refer to its two children as the "zero" child and the "one" child. A random Quadtree $\mathbf{T}_0$ is drawn by sampling a coordinate $\boldsymbol{i} \sim [d]$ uniformly and independently for each internal node of depth at most d as its label; every internal node of depth $d + j$, $j \in [d]$, is labelled j.*

Given a Quadtree T_0, each point $x \in \{0,1\}^d$ can be assigned to a leaf of T_0 by starting at the root and repeatedly going down to the "zero" child if $x_i = 0$ for the label i of the current node, or to the "one" child if $x_i = 1$. Alternatively one can define a subcube $S_v \subseteq \{0,1\}^d$ for each node v: The set for the root is $\{0,1\}^d$; If (u, v) is an edge, u is labelled i, and v is the "zero" (or "one") child of u, then $S_v = \{y \in S_u : x_i = 0\}$ (or $S_v = \{y \in S_u : x_i = 1\}$). Each point $x \in \{0,1\}^d$ is then assigned to the unique leaf v with $x \in S_v$. Given a Quadtree T_0, we write A_v to denote $A \cap S_v$ and B_v to denote $B \cap S_v$ for each node v of T_0 (thus A_v contains all points $a \in A$ such that v is on its root-to-leaf path in T_0, and the same holds for B_v).

Next we define the compressed version of a Quadtree. Roughly speaking, we compress a Quadtree T_0 by contracting all nodes of T_0 at depths that are not powers of 2, and keeping every other node as well as its subcube S_v (including the pair of multi-sets A_v and B_v).

Definition 2.2 (Compression)**.** *Given any Quadtree T_0 of depth $2d$, we write $T = \textsc{Compress}(T_0)$ to denote the following rooted tree of depth $h = \log 2d$. For each $i = 1, \ldots, h$, there is a one-to-one*

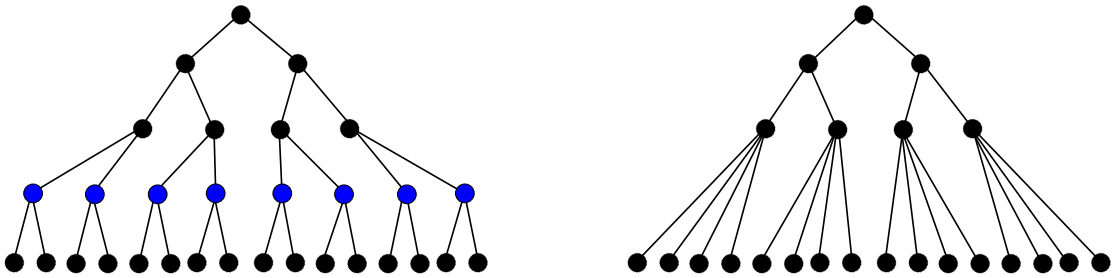

Figure 2: The Compression Operation. The Quadtree T_0 on the left-hand side is a complete binary tree, and its $\textsc{Compress}(T_0)$ is represented on the right-hand side. The compression contracts nodes which are not at a depth that is a power of 2, which in this case, corresponds to nodes at depth 3 (labeled in blue). Note that every node of T naturally corresponds with a node of T_0 (in particular, we have displayed nodes as remaining in the same relative position to illustrate this point).

correspondence between nodes of T at depth i and nodes of T_0 at depth 2^i; the root of T corresponds to the root of T_0. Node v is a child of u in T if the corresponding node of v is a descendant of the corresponding node of u in T_0. Each node v in T is labelled a subcube S_v and a pair of multi-sets A_v and B_v, copied from its corresponding node in T_0. See Figure 2 for an example of $\textsc{Compress}(T_0)$. *A random compressed Quadtree $\mathbf{T}$ is drawn by first drawing a random Quadtree $\mathbf{T}_0$ and then setting $\mathbf{T} = \textsc{Compress}(\mathbf{T}_0)$.*

The key property we use about a random compressed Quadtree is summarized in the lemma below (which follows directly from its definition):

Lemma 2.3. *Fix any point $x \in \{0,1\}^d$. Let $\mathbf{T}$ be a random compressed Quadtree and $\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_h$ be the root-to-leaf path of x in $\mathbf{T}$. Then $(S_{\mathbf{v}_1}, \ldots, S_{\mathbf{v}_{h-1}})$ can be drawn equivalently as follows[15]: First draw $\mathbf{I}_0, \mathbf{I}_1, \ldots, \mathbf{I}_{h-2}$ where $\mathbf{I}_0, \mathbf{I}_1 \sim [d]$ and $\mathbf{I}_i \sim [d]^{2^{i-1}}$ for each $i \geq 1$ independently and uniformly at random, and set $S_{\mathbf{v}_i}$ to be the subcube of $\{0,1\}^d$ that agrees with x on all coordinates in $\mathbf{I}_0, \ldots, \mathbf{I}_{i-1}$.*

Remark 4. *Both works of* [AIK08, BDI+20] *use a tree structure that is very similar to a compressed Quadtree as defined above. They consider a slightly different divide-and-conquer algorithm which at depth i, samples 2^i coordinates from $[d]$ and divides into 2^{2^i} branches according to settings of $\{0,1\}$ to these 2^i coordinates. The resulting tree structure is akin to a compressed Quadtree, in which each node at depth i has 2^{2^i} children. As we will see, our analysis apply to trees of* [AIK08, BDI+20] *as well given that the lemma above also holds trivially for their trees.*

3 Analysis of Compressed Quadtrees

Let A, B be multi-sets of $\{0,1\}^d$ of size s, and let $M^* \subset A \times B$ be an optimal matching so that

$$\mathrm{EMD}(A,B) = \sum_{(a,b)\in M^*} \|a-b\|_1.$$

[15] Note that $S_{\mathbf{v}_0}$ is trivially $\{0,1\}^d$ and $S_{\mathbf{v}_h}$ is trivially $\{x\}$.

We prove Theorem 1 in this section. For the time complexity of ComputeEMD(A, B, d), we note that the running time of each call with $h > 1$ (excluding running time from its recursive calls) is linear in $|A| + |B|$, the total size of its two input sets. On the other hand, the running time of each call with $h = 0$ can be bounded by $O((|A| + |B|)d)$. (Recall that we need to find a maximal partial matching M that matches as many identical pairs of points of A and B as possible. This can be done by first sorting $A \cup B$ and noting that given there are only 2^d points in the space, we only need to pay $O(d)$ for each insertion instead of $O(\log(|A| + |B|))$ which could be larger than d.) Equivalently, one can charge $O(1)$ running time to each point in the two input sets of an internal node and $O(d)$ to each point at each leaf of the recursion tree. Therefore, each point pays at most $O(d)$ given that ComputeEMD has depth at most d. It follows that its overall running time is $O(sd)$. Given the cost of any matching is trivially at least $\text{EMD}(A, B)$, it suffices to upperbound the cost of the matching returned by ComputeEMD by $\tilde{O}(\log s) \cdot \text{EMD}(A, B)$ with probability at least 0.9, which we focus on in the rest of the section.

To this end, we first note that ComputeEMD can be viewed as first drawing a random compressed Quadtree $\mathbf{T}$ and then returning a matching $\mathbf{M}$ built in the bottom-up fashion as described in Figure 1. Given a fixed compressed Quadtree T, the following definition captures the class of matchings that ComputeEMD may return; basically these are matchings that satisfy a "depth-greedy" property:

Definition 3.1. *Let T be a compressed Quadtree. For any $a \in A$ and $b \in B$, let*

$$\textit{depth}_T(a, b) \stackrel{\text{def}}{=} \textit{depth of the least-common ancestor of leaves of } a, b \textit{ in } T.$$

The class of depth-greedy matchings*, denoted by $\mathcal{M}_T(A, B)$, is the set of all matchings $M \subseteq A \times B$ which maximize the sum of $\textit{depth}_T(a, b)$ over all pairs $(a, b) \in M$.*

Fixing a compressed Quadtree T, every matching ComputeEMD may return when running with T belongs to $\mathcal{M}_T(A, B)$ (indeed $\mathcal{M}_T(A, B)$ can contain strictly more matchings in general). Therefore the goal is now to show that with probability at least 0.9 over $\mathbf{T}$, $\mathbf{Cost}(M)$ of every $M \in \mathcal{M}_{\mathbf{T}}(A, B)$ is upperbounded by $\tilde{O}(\log s) \cdot \text{EMD}(A, B)$.

Our plan consists of three steps. First we introduce the *cost* of a compressed Quadtree T (Definition 3.2 below), denoted by $\mathbf{Cost}(T)$, and show that $\mathbf{Cost}(M)$ of any matching $M \in \mathcal{M}_T(A, B)$ can be bounded from above by $\mathbf{Cost}(T)$ (Lemma 3.3). (Looking ahead, $\mathbf{Cost}(T)$ will be the quantity our linear sketches estimate in Sections 5 and 6; see Remark 5.) Next we define payments $\mathbf{Pay}_T(a)$ and $\mathbf{Pay}_T(b)$ of the inspector as mentioned in the overview earlier, and show that $\mathbf{Cost}(T)$ is bounded from above by the total inspector payment (Lemma 3.4). The final and most challenging step is to bound the expected total inspector payment. The key technical lemma, Lemma 3.5, will be proved in Section 4.

We start with the cost of a compressed Quadtree. We need a couple of definitions used throughout the rest of the paper. Given a compressed Quadtree T and a point $x \in \{0, 1\}^d$, we let the sequence

$$v_{0,T}(x), v_{1,T}(x), \ldots, v_{h,T}(x)$$

denote the root-to-leaf path of x in the tree T. For any node v at depth $i \in \{0, \ldots, h\}$, we let

$$A_{v,T} \stackrel{\text{def}}{=} \{a \in A : v_{i,T} = v\} \qquad \text{and} \qquad B_{v,T} \stackrel{\text{def}}{=} \{b \in B : v_{i,T} = v\},$$

and we let

$$c_{v,T} \stackrel{\text{def}}{=} \frac{1}{|A_{v,T} \cup B_{v,T}|} \sum_{x \in A_{v,T} \cup B_{v,T}} x$$

be the *center-of-mass* of points in $A_{v,T} \cup B_{v,T}$ at v; we set $c_{v,T}$ to be the all-0 point by default when $A_{v,T} = B_{v,T} = \emptyset$. For notational simplicity, we will usually suppress T from the notation when it is clear from context (this will also be the case for notation introduced later in this section). We are now ready to define the cost of a compressed Quadtree:

Definition 3.2. *Let T be a compressed Quadtree. Then its cost is defined as*[16]

$$\mathbf{Cost}(T) \stackrel{\text{def}}{=} \sum_{(u,v) \in E_T} \big||A_v| - |B_v|\big| \cdot \|c_v - c_u\|_1. \tag{12}$$

Notice that the right-hand side of (12) is data-dependent in two respects: 1) each edge of the tree contributes the discrepancy between A and B proceeding down that edge, and 2) the amount each edge pays times the discrepancy between A and B is given by the distance between centers-of-mass of the point set of $A \cup B$ mapped to u and v.

We prove the following lemma for the first step of the proof.

Lemma 3.3. *Let T be a compressed Quadtree. Then* $\mathbf{Cost}(M) \leq \mathbf{Cost}(T)$ *for all $M \in \mathcal{M}_T(A, B)$.*

Proof: Given a matching $M \in \mathcal{M}_T(A, B)$ and a pair $(a, b) \in M$, we write v and w to denote the leaves of a and b and use u to denote their least-common ancestor. We also write $u, v_1, \ldots, v_k = v$ and $u, w_1, \ldots, w_k = w$ to denote the paths from u to v and w, respectively. By triangle inequality,

$$\begin{aligned}\|a - b\|_1 \leq{}& \|a - c_{v_k}\|_1 + \|c_{v_k} - c_{v_{k-1}}\|_1 + \cdots + \|c_{v_1} - c_u\|_1 \\ &+ \|c_u - c_{w_1}\| + \cdots + \|c_{w_{k-1}} - c_{w_k}\|_1 + \|c_{w_k} - b\|_1 \\ ={}& \|c_{v_k} - c_{v_{k-1}}\|_1 + \cdots + \|c_{v_1} - c_u\|_1 + \|c_u - c_{w_1}\| + \cdots + \|c_{w_{k-1}} - c_{w_k}\|_1,\end{aligned}$$

where the equation follows from the fact that all points at a leaf of T must be identical and so is their center. Summing up these inequalities over $(a, b) \in M$ gives exactly $\mathbf{Cost}(T)$. For this, note that every M in $\mathcal{M}_T(A, B)$ has the property that, for any edge (u, v) in T, the number of $(a, b) \in M$ such that the path between their leaves contains (u, v) is exactly $||A_v| - |B_v||$. ■

Now it suffices to upperbound $\mathbf{Cost}(\mathbf{T})$ by $\tilde{O}(\log s) \cdot \mathrm{EMD}(A, B)$ with probability at least 0.9 for a random compressed Quadtree $\mathbf{T}$. For this purpose, we define an inspector payment for each point in $A \cup B$ based on an optimal matching M^* between A and B and a compressed Quadtree T. Given a point $a \in A$, if $b \in B$ is the point matched with a in M^* (i.e., $(a, b) \in M^*$), we let

$$\mathbf{Pay}_T(a) \stackrel{\text{def}}{=} \sum_{i \in [h]} \mathbf{1}\{v_{i,T}(a) \neq v_{i,T}(b)\} \Big(\|a - c_{v_{i,T}(a)})\|_1 + \|a - c_{v_{i-1,T}(a)}\|_1 \Big). \tag{13}$$

Intuitively $\mathbf{Pay}_T(a)$ pays for the distance between a and centers-of-mass along its root-to-leaf path but the payment only starts *at* the least-common ancestor of leaves of a, b. The payment $\mathbf{Pay}_T(b)$ is defined similarly (with a and b switched). Note that $\mathbf{Pay}_T(a) = \mathbf{Pay}_T(b) = 0$ if $a = b$.

We show that the total inspector payment from points of $A \cup B$ is enough to cover $\mathbf{Cost}(T)$:

[16] Whenever we refer to an edge (u, v) in T, u is always the parent node and v is the child node.

Lemma 3.4. *Let T be any compressed Quadtree. Then we have*

$$\mathbf{Cost}(T) \leq \sum_{a\in A} \mathbf{Pay}_T(a) + \sum_{b\in B} \mathbf{Pay}_T(b). \tag{14}$$

Proof: Using the definition of $\mathbf{Cost}(T)$, it suffices to show that

$$\sum_{(u,v)\in E_T} \big||A_v| - |B_v|\big| \cdot \|c_v - c_u\|_1 \leq \sum_{a\in A} \mathbf{Pay}_T(a) + \sum_{b\in B} \mathbf{Pay}_T(b).$$

By triangle inequality,

$$\mathbf{Pay}_T(a) \geq \sum_{i\in[h]} \mathbf{1}\{v_{i,T}(a) \neq v_{i,T}(b)\}\Big(\|c_{v_{i-1,T}}(a) - c_{v_{i,T}(a)})\|_1\Big),$$

i.e., $\mathbf{Pay}_T(a)$ is enough to cover $\|c_u - c_v\|$ for every edge (u,v) along its leaf-to-root path until the least-common ancestor with b is met. The lemma then follows from the following claim: For every edge (u,v) in T, $||A_v| - |B_v||$ is at most the number of points $a \in A_v$ such that its matched point in M^* is not in B_v plus the number of points $b \in B_v$ such that its matched point in M^* is not in A_v. This follows from the simple fact that every $(a,b) \in M^*$ with $a \in A_v$ and $b \in B_v$ would get cancelled in $|A_v| - |B_v|$. This finishes the proof of the lemma. ■

By Lemma 3.4 the goal now is to upperbound the total inspector payment by $\tilde{O}(\log s) \cdot \mathrm{EMD}(A,B)$ with probability at least 0.9 over a randomly picked compressed Quadtree $\mathbf{T}$. We consider a slight modification of the payment scheme given in (13) which we define next; the purpose is that the latter will be easier to bound in expectation, and most often exactly equal to (13).

Specifically, given a pair $(a,b) \in M^*$ and $i_0 \in [h]$, we let $\widetilde{\mathbf{Pay}}_{i_0,T}(a) = 0$ if $a = b$ and let

$$\widetilde{\mathbf{Pay}}_{i_0,T}(a) \stackrel{\text{def}}{=} \sum_{i=i_0}^{h} \mathbf{1}\{v_{i,T}(a) \neq v_{i,T}(b)\}\Big(\|a - \tilde{c}_T(i,a))\|_1 + \|a - \tilde{c}_T(i-1,a)\|_1\Big) \tag{15}$$

when $a \neq b$, where

$$\tilde{c}_T(i,a) \stackrel{\text{def}}{=} \frac{1}{|C_T(i,a)|} \sum_{x\in C_T(i,a)} x$$

is the center-of-mass of a subset $C_T(i,a)$ of $A_{v_{i,T}(a)} \cup B_{v_{i,T}(a)}$ that are not too far away from a:

$$C_T(i,a) \stackrel{\text{def}}{=} \left\{ x \in A_{v_{i,T}(a)} \cup B_{v_{i,T}(a)} : \|x - a\|_1 \leq \frac{10d\log s}{2^i} \right\}.$$

Roughly speaking, two points in $A \cup B$ that share the same node $v_{i,T}$ are expected to have distance around $d/2^i$ (given that they agreed so far on roughly 2^i random coordinates sampled); this is why we referred to points in $C_T(i,a)$ as those that are not too far away from a. Similar to $\mathbf{Pay}_T(a)$, we define $C_T(i,b)$ and $\tilde{c}_T(i,b)$ for each $b \in B$ and use them to define $\widetilde{\mathbf{Pay}}_T(b)$.

The following is the crucial lemma for bounding the total expected payment from points in $A \cup B$. We delay its proof to Section 3.5 and first use it to prove Theorem 1.

Lemma 3.5. *For any $(a,b) \in M^*$ with $a \neq b$ and $i_0 \in [h]$ that satisfies*

$$i_0 \leq \min\left\{1, \left\lfloor \log_2\left(\frac{d}{\|a-b\|_1}\right)\right\rfloor\right\}, \tag{16}$$

we have

$$\max\left\{\mathop{\mathbf{E}}_{\mathbf{T}}\left[\widetilde{\mathbf{Pay}}_{i_0,\mathbf{T}}(a)\right], \mathop{\mathbf{E}}_{\mathbf{T}}\left[\widetilde{\mathbf{Pay}}_{i_0,\mathbf{T}}(b)\right]\right\} \leq \left(\tilde{O}(\log s) + O(\log\log s)\left(\log\left(\frac{d}{\|a-b\|_1}\right) - i_0\right)\right)\cdot\|a-b\|_1,$$

where the randomness is over a random compressed Quadtree $\mathbf{T}$.

Proof of Theorem 1 assuming Lemma 3.5: Let $\mathbf{T}$ be a random compressed Quadtree. Then

$$\mathbf{Cost}(\mathbf{T}) \leq \sum_{a\in A}\mathbf{Pay}_{\mathbf{T}}(a) + \sum_{b\in B}\mathbf{Pay}_{\mathbf{T}}(b) = \sum_{\substack{(a,b)\in M^*\\ a\neq b}}\mathbf{Pay}_{\mathbf{T}}(a) + \mathbf{Pay}_{\mathbf{T}}(b) \tag{17}$$

given that $\mathbf{Pay}_{\mathbf{T}}(a) = \mathbf{Pay}_{\mathbf{T}}(b) = 0$ for every pair $(a,b) \in M^*$ with $a = b$. We focus on the subset M' of M^* with $(a,b) \in M^*$ and $a \neq b$. For each pair $(a,b) \in M'$, let

$$\ell_{\min}(a,b) \stackrel{\text{def}}{=} \min\left\{1, \left\lfloor \log_2\left(\frac{d}{\|a-b\|_1}\right)\right\rfloor - 2\lceil\log_2 s\rceil\right\}.$$

We show that with probability at least $1 - o(1)$ over the draw of $\mathbf{T}$, every $(a,b) \in M'$ satisfies

$$\mathbf{Pay}_{\mathbf{T}}(a) = \widetilde{\mathbf{Pay}}_{\ell_{\min}(a,b),\mathbf{T}}(a) \quad \text{and} \quad \mathbf{Pay}_{\mathbf{T}}(b) = \widetilde{\mathbf{Pay}}_{\ell_{\min}(a,b),\mathbf{T}}(b). \tag{18}$$

Combining (17) and (18), we have that with probability at least $1 - o(1)$ over the draw of $\mathbf{T}$,

$$\mathbf{Cost}(\mathbf{T}) \leq \sum_{(a,b)\in M'}\widetilde{\mathbf{Pay}}_{\ell_{\min}(a,b),\mathbf{T}}(a) + \widetilde{\mathbf{Pay}}_{\ell_{\min}(a,b),\mathbf{T}}(b). \tag{19}$$

By applying Lemma 3.5 to every $(a,b) \in M'$ with $i_0 = \ell_{\min}(a,b)$, as well as Markov's inequality, we have that with probability at least 0.99 over $\mathbf{T}$, the right hand side of (19) is at most

$$\tilde{O}(\log s)\sum_{(a,b)\in M'}\|a-b\|_1 = \tilde{O}(\log s)\cdot\mathrm{EMD}(A,B).$$

By a union bound, $\mathbf{Cost}(\mathbf{T}) \leq \tilde{O}(\log s)\cdot\mathrm{EMD}(A,B)$ with probability at least $.99 - o(1) \geq 0.9$.

It suffices to define an event that implies (18) and then bound its probability. The first part of the event requires that for every pair $(a,b) \in M'$, $v_{i,\mathbf{T}}(a) = v_{i,\mathbf{T}}(b)$ for every $i : 1 \leq i < \ell_{\min}(a,b)$. The second part requires that for any two distinct points $x, y \in A \cup B$ (not necessarily as a pair in M^* and not even necessarily in the same set), we have $v_{i,\mathbf{T}}(x) \neq v_{i,\mathbf{T}}(y)$ for all i with

$$2^i \geq \frac{10d\log s}{\|x-y\|_1}. \tag{20}$$

By the definition of $\widetilde{\mathbf{Pay}}_{\ell_{\min}(a,b),\mathbf{T}}(a)$ in (15) the first part of the event makes sure that we don't miss any term in the sum (15); the second part of the event makes sure that every $C_{\mathbf{T}}(i,a)$ is exactly the same as $A_{v_{i,\mathbf{T}}}(a) \cup B_{v_{i,\mathbf{T}}}(a)$ (and the same holds for b). It follows that this event implies (18).

Finally we show that the event occurs with probability at least $1-o(1)$. First, for every $(a,b) \in M'$, the probability of $v_{i,\mathbf{T}}(a) \neq v_{i,\mathbf{T}}(b)$ for some $i : 1 \leq i < \ell_{\min}(a,b)$ is at most

$$1 - \left(1 - \frac{\|a-b\|_1}{d}\right)^{2^{\ell_{\min}(a,b)}} \leq 2^{\ell_{\min}(a,b)} \cdot \frac{\|a-b\|_1}{d} \leq \frac{1}{s^2}.$$

Hence, by a union bound over the at most s pairs $(a,b) \in M'$, the first part of the event holds with probability at least $1-o(1)$. Furthermore, for any two distinct points $x, y \in A \cup B$, let

$$\ell_{\max}(x,y) = \left\lfloor \log_2 \left(\frac{10d \log s}{\|x-y\|_1} \right) \right\rfloor.$$

Then $v_{i,\mathbf{T}}(x) = v_{i,\mathbf{T}}(y)$ for some i that satisfies (20) would imply $v_{\ell_{\max}(x,y),\mathbf{T}}(x) = v_{\ell_{\max}(x,y),\mathbf{T}}(y)$ and $\ell_{\max}(x,y) \leq \log d$ (as $v_{h,\mathbf{T}}(x) \neq v_{h,\mathbf{T}}(y)$ given $x \neq y$). The event above happens with probability

$$\left(1 - \frac{\|x-y\|_1}{d}\right)^{2^{\ell_{\max}(x,y)-1}} \leq \exp(-5 \log s) = \frac{1}{s^5}.$$

Via a union bound over at most $(2s)^2$ many pairs of x, y, we have that the second part of the event also happens with probability at least $1-o(1)$. This finishes the proof of the theorem. ■

Remark 5 (Looking toward Section 5 and 6)**.** *Before moving on to the proof of Lemma 4 we define a quantity that is slightly different from* $\mathbf{Cost}(T)$ *which will be used in our linear sketches of Section 5 and 6 to estimate* $\mathrm{EMD}(A,B)$*.*

An inspection of the proof of Theorem 1 reveals that with probability $1-o(1)$ *over* $\mathbf{T}$ *(which we skip in subscripts below), every non-empty node* v *(*$A_v \cup B_v \neq \emptyset$*) at depth* i *with parent* u *satisfies*

$$\|\mathbf{c}_v - \mathbf{c}_u\|_1 \leq \frac{30d \log s}{2^i}.$$

The reason is that we have argued, with probability at least $1-o(1)$ *over* $\mathbf{T}$*, for all* $x \in A \cup B$ *and all depth* i*, all points in* $A_{v_i(x)} \cup B_{v_i(x)}$ *are within distance* $10d \log s / 2^i$ *of* x*. This implies that every non-empty node* v *(say* $x \in A_v \cup B_v$*) at depth* i *and its parent* u *satisfy*

$$\|\mathbf{c}_v - \mathbf{c}_u\|_1 \leq \|\mathbf{c}_v - x\|_1 + \|\mathbf{c}_u - x\|_1 \leq \frac{10d \log s}{2^i} + \frac{10d \log s}{2^{i-1}} = \frac{30d \log s}{2^i}.$$

In particular, with probability at least 0.9 *over* $\mathbf{T}$*, we have*

$$\mathrm{EMD}(A,B) \leq \sum_{(u,v) \in E_{\mathbf{T}}} \big||A_v| - |B_v|\big| \cdot \min\left\{ \|\mathbf{c}_v - \mathbf{c}_u\|_1, \frac{30d \log s}{2^i} \right\} \leq \tilde{O}(\log s) \cdot \mathrm{EMD}(A,B). \quad (21)$$

Furthermore, recall from the embedding into ℓ_1*, we have with probability at least* 0.99 *over* $\mathbf{T}$*,*

$$\sum_{(u,v) \in E_{\mathbf{T}}} \big||A_v| - |B_v|\big| \cdot \frac{d}{2^i} \leq O(\log s) \cdot \mathrm{EMD}(A,B). \quad (22)$$

This follows from an analysis similar to (3) (although (3) only gives $O(\log d)$ on the right hand side instead of the $O(\log s)$ we need, one can improve it to $\min(\log s, \log d)$; see footnote 6 and an example of implementing this idea in Lemma A.2).

Combining (21) and (22), we have that with probability at least 0.9 over the draw of $\mathbf{T}$,

$$\mathrm{EMD}(A,B) \le \sum_{(u,v)\in E_{\mathbf{T}}} \big||A_v| - |B_v|\big| \cdot \boldsymbol{q}_{u,v} \le \tilde{O}(\log s)\cdot \mathrm{EMD}(A,B), \tag{23}$$

where $\boldsymbol{q}_{u,v}$ is defined as the following truncation of $\|\mathbf{c}_u - \mathbf{c}_v\|_1$:

$$\boldsymbol{q}_{u,v} \stackrel{\text{def}}{=} \begin{cases} d/2^i & \text{if } \|\mathbf{c}_u - \mathbf{c}_v\|_1 \le d/2^i \\ 30d\log s/2^i & \text{if } \|\mathbf{c}_u - \mathbf{c}_v\|_1 \ge 30d\log s/2^i \\ \|\mathbf{c}_u - \mathbf{c}_v\|_1 & \text{otherwise.} \end{cases}$$

The sum in the center of (23) will be crucial in later sections; this will be the quantity we estimate in our linear sketches in Section 5 and 6.

4 Proof of Lemma 3.5

Recall $h = \log 2d$ is the depth of a compressed Quadtree. Let $(a,b) \in M^*$ with $a \ne b$ and $i_0 \in [h]$ that satisfy (16). We need to bound the expectation of $\widetilde{\mathbf{Pay}}_{i_0}(a)$ over the draw of a random compressed Quadtree $\mathbf{T}$; the upper bound for $\widetilde{\mathbf{Pay}}_{i_0}(b)$ is analogous. In what follows, all expectations are taken with respect to a random choice of $\mathbf{T}$ so we skip $\mathbf{T}$ in subscripts (just as in $\widetilde{\mathbf{Pay}}_{i_0}(a)$). In particular, we write $\mathbf{v}_i(a)$ to denote $v_{i,\mathbf{T}}(a)$ just to emphasize that it is a random variable that depends on $\mathbf{T}$.

Given that we always have $\|a - \widetilde{\mathbf{c}}(i,a)\|_1 \le 10d\log s/2^i$ by definition, $\widetilde{\mathbf{Pay}}_{i_0}(a) = O(d\log s)$; hence, we assume that $\|a-b\|_1 \le d/2$, since otherwise the lemma is trivially true. To understand $\widetilde{\mathbf{Pay}}_{i_0}(a)$ for a random compressed Quadtree $\mathbf{T}$, we need to examine $A_{\mathbf{v}_i(a)}$ and $B_{\mathbf{v}_i(a)}$ where $\mathbf{v}_0(a), \dots, \mathbf{v}_h(a)$ is the root-to-leaf path of a in $\mathbf{T}$. Let $\tau_0 = 1$ and $\tau_i = 2^{i-1}$ when $i \ge 1$. Recall from Lemma 2.3 that to draw $A_{\mathbf{v}_i(a)}$ and $B_{\mathbf{v}_i(a)}$, it suffices to consider independent draws

$$\mathbf{I}_0, \mathbf{I}_1, \dots, \mathbf{I}_{h-2}$$

with $\mathbf{I}_i \sim [d]^{\tau_i}$, and then use them to define $A_{\mathbf{v}_i(a)}$ and $B_{\mathbf{v}_i(a)}$ as follows:

$$A_{\mathbf{v}_i(a)} = \{x \in A : x_j = a_j \text{ for all } j \text{ that appears in } \mathbf{I}_0, \dots, \mathbf{I}_{i-1}\}$$

$$B_{\mathbf{v}_i(a)} = \{x \in B : x_j = a_j \text{ for all } j \text{ that appears in } \mathbf{I}_0, \dots, \mathbf{I}_{i-1}\}$$

for each $i = 1, \dots, h-1$; $A_{\mathbf{v}_0(a)} = A$ and $B_{\mathbf{v}_0(a)} = B$ since $\mathbf{v}_0(a)$ is always the root; $A_{\mathbf{v}_h(a)}$ contains all copies of a in the multi-set A and $B_{v_h(b)}$ contains all copies of a in B. We let

$$\mathbf{D} = \big\{(i,\ell) : i \in \{0,\dots,h-2\} \text{ and } \ell \in [\tau_i] \text{ such that } j = (\mathbf{I}_i)_\ell \text{ satisfies } a_j \ne b_j\big\}$$

be the set of index pairs of sampled coordinates where a and b disagree, and let

$$(\boldsymbol{i}^{(s)}, \boldsymbol{\ell}^{(s)}) \stackrel{\text{def}}{=} \begin{cases} \min \mathbf{D} & \mathbf{D} \ne \emptyset \\ (*,*) & \mathbf{D} = \emptyset \end{cases},$$

where the ordering in $\min \mathbf{D}$ is lexicographic.[17] Note, in particular, that the node $\mathbf{v}_{\boldsymbol{i}^{(s)}}(a) = \mathbf{v}_{\boldsymbol{i}^{(s)}}(b)$

[17] All (i',ℓ') with $i' < \boldsymbol{i}^{(s)}$ satisfy $a_j = b_j$ for $j = (\mathbf{I}_{i'})_{\ell'}$; all $(\boldsymbol{i}^{(s)}, \ell')$ with $\ell' < \boldsymbol{\ell}^{(s)}$ satisfy $a_j = b_j$ for $j = (\mathbf{I}_{\boldsymbol{i}^{(s)}})_{\ell'}$.

is the least common ancestor of a and b in $\mathbf{T}$.[18] The coordinate which is the first to satisfy $a_j \neq b_j$ is specified by the random variable

$$\boldsymbol{j}^{(s)} \stackrel{\text{def}}{=} \begin{cases} (\mathbf{I}_{\boldsymbol{i}^{(s)}})_{\boldsymbol{\ell}^{(s)}} & \text{if } (\boldsymbol{i}^{(s)}, \boldsymbol{\ell}^{(s)}) \neq (*,*) \\ * & \text{otherwise} \end{cases},$$

where, notice that, $\boldsymbol{j}^{(s)} = *$ only if a and b always satisfy $\mathbf{v}_k(a) = \mathbf{v}_k(b)$ for all $k \in \{0, \ldots, h-1\}$. A trivial consequence of the above definitions is that for any $k \in \{0, \ldots, h-1\}$,

$$\mathbf{1}\{\mathbf{v}_k(a) \neq \mathbf{v}_k(b)\} = \sum_{i=0}^{k-1} \sum_{\ell=1}^{\tau_i} \sum_{\substack{j \in [d] \\ a_j \neq b_j}} \mathbf{1}\left\{(\boldsymbol{i}^{(s)}, \boldsymbol{\ell}^{(s)}, \boldsymbol{j}^{(s)}) = (i, \ell, j)\right\}. \tag{24}$$

From the definition of $\widetilde{\mathbf{Pay}}_{i_0}(a)$ in (15), letting

$$h_{a,b} \stackrel{\text{def}}{=} \left\lfloor \log_2 \left(\frac{d}{\|a-b\|_1} \right) \right\rfloor \leq \log_2 d = h - 1$$

and we have

$$\mathbf{E}\left[\widetilde{\mathbf{Pay}}_{i_0}(a)\right] \leq \sum_{k=i_0}^{h_{a,b}} \mathbf{E}\left[\mathbf{1}\{\mathbf{v}_k(a) \neq \mathbf{v}_k(b)\}\left(\|a - \widetilde{\mathbf{c}}(k,a)\|_1 + \|a - \widetilde{\mathbf{c}}(k-1,a)\|_1\right)\right] + O(\log s) \cdot \|a-b\|_1,$$

where we used the fact that $\|a - \widetilde{\mathbf{c}}(k,a)\|_1 \leq 10d \log s / 2^k$ always holds. As a result $O(\log s) \cdot \|a-b\|_1$ as in the last term is enough to cover the sum over $k > h_{a,b}$ skipped in the above expression. Using (24), we may re-write the first summand above as

$$\begin{aligned} &\sum_{k=i_0}^{h_{a,b}} \mathbf{E}\left[\mathbf{1}\{\mathbf{v}_k(a) \neq \mathbf{v}_k(b)\}\left(\|a - \widetilde{\mathbf{c}}(k,a)\|_1 + \|a - \widetilde{\mathbf{c}}(k-1,a)\|_1\right)\right] \qquad (25) \\ &= \sum_{k=i_0}^{h_{a,b}} \sum_{i=0}^{k-1} \sum_{\ell=1}^{\tau_i} \sum_{\substack{j \in [d] \\ a_j \neq b_j}} \mathbf{Pr}\left[(\boldsymbol{i}^{(s)}, \boldsymbol{\ell}^{(s)}, \boldsymbol{j}^{(s)}) = (i,\ell,j)\right] \Bigg(\mathbf{E}\left[\|a - \widetilde{\mathbf{c}}(k,a)\|_1 \,\middle|\, (\boldsymbol{i}^{(s)}, \boldsymbol{\ell}^{(s)}, \boldsymbol{j}^{(s)}) = (i,\ell,j)\right] \\ &\qquad\qquad + \mathbf{E}\left[\|a - \widetilde{\mathbf{c}}(k-1,a)\|_1 \,\middle|\, (\boldsymbol{i}^{(s)}, \boldsymbol{\ell}^{(s)}, \boldsymbol{j}^{(s)}) = (i,\ell,j)\right] \Bigg). \end{aligned}$$

Notice that for $i \in \{0, \ldots, h-2\}$, $\ell \in [\tau_i]$, and $j \in [d]$ with $a_j \neq b_j$,

$$\mathbf{Pr}\left[(\boldsymbol{i}^{(s)}, \boldsymbol{\ell}^{(s)}, \boldsymbol{j}^{(s)}) = (i,\ell,j)\right] = \frac{1}{d}\left(1 - \frac{\|a-b\|_1}{d}\right)^{\tau_0 + \cdots + \tau_{i-1} + \ell - 1}. \tag{26}$$

Consider, for each $k \in \{i_0 - 1, \ldots, h_{a,b}\}$ and $j \in [d]$ with $a_j \neq b_j$, the value

$$Q_{k,j} \stackrel{\text{def}}{=} \sup_{\substack{i \in \{0,\ldots,k\} \\ \ell \in [\tau_i]}} \mathbf{E}\left[\|a - \widetilde{\mathbf{c}}(k,a)\|_1 \,\middle|\, (\boldsymbol{i}^{(s)}, \boldsymbol{\ell}^{(s)}, \boldsymbol{j}^{(s)}) = (i,\ell,j)\right],$$

[18] Hence, the "s" in $\boldsymbol{i}^{(s)}$ and $\boldsymbol{\ell}^{(s)}$ stands for "split."

so that invoking (26), we may upper bound (25) by

$$\sum_{\substack{j\in[d]\\ a_j\neq b_j}}\sum_{k=i_0}^{h_{a,b}}\left(Q_{k,j}+Q_{k-1,j}\right)\sum_{i=0}^{k-1}\sum_{\ell=1}^{\tau_i}\frac{1}{d}\left(1-\frac{\|a-b\|_1}{d}\right)^{\tau_0+\cdots+\tau_{i-1}+\ell-1}\leq\frac{4}{d}\sum_{\substack{j\in[d]\\ a_j\neq b_j}}\sum_{k=i_0-1}^{h_{a,b}}Q_{k,j}\cdot 2^k.$$

Therefore, it remains to show that for all $j\in[d]$ with $a_j\neq b_j$,

$$\sum_{k=i_0-1}^{h_{a,b}}Q_{k,j}\cdot 2^k=O(d)\cdot\left(\tilde{O}(\log s)+(h_{a,b}-i_0)\cdot\log\log s\right). \tag{27}$$

To prove this, we start with a lemma that shows that if $Q_{k,j}$ is large, then it must be the case that there are many points of distance between (roughly) $d/(2^k\log s)$ and $d\log s/2^k$ from a in $A\cup B$.

Lemma 4.1. *Fix $j\in[d]$ which satisfies $a_j\neq b_j$ and $k\in\{2,\ldots,h_{a,b}\}$. Let $L\subset A_j\cup B_j\subset A\cup B$ be the muti-sets given by*

$$A_j=\{x\in A: x_j=a_j\},\quad B_j=\{x\in B: x_j=a_j\}\quad \textit{and}$$
$$L=\left\{x\in A_j\cup B_j:\|a-x\|_1\leq\frac{d}{2^k}\cdot\frac{1}{\log s}\right\}.$$

Suppose that for some $i=\{0,\ldots,k\}$ and some $\ell\in[\tau_i]$, as well as some $\alpha\geq 300$, we have

$$\mathbf{E}\left[\|a-\widetilde{\mathbf{c}}(k,a)\|_1\,\middle|\,(\boldsymbol{i}^{(s)},\boldsymbol{\ell}^{(s)},\boldsymbol{j}^{(s)})=(i,\ell,j)\right]\geq\frac{d}{2^k}\cdot\alpha. \tag{28}$$

Then, the set

$$H=\left\{x\in A_j\cup B_j:\|a-x\|_1\leq\frac{10d\log s}{2^k}\right\}\quad \textit{satisfies}\quad |H|\geq\frac{|L|\exp\left(\alpha/8\right)}{\log s}.$$

Proof: Let $\boldsymbol{\mathcal{E}}$ be the event that $(\boldsymbol{i}^{(s)},\boldsymbol{\ell}^{(s)},\boldsymbol{j}^{(s)})=(i,\ell,j)$. For simplicity in the notation, let

$$\mathbf{v}=\mathbf{v}_k(a),\quad \mathbf{C}=\mathbf{C}(k,a)\quad\text{and}\quad \mathbf{c}=\widetilde{\mathbf{c}}(k,a).$$

Every $x\in L$ is of distance at most $d/(2^k\log s)$ from a. Since $x\in\mathbf{C}$ whenever $\mathbf{v}_k(x)=\mathbf{v}$, we have

$$\mathbf{E}\left[|L\setminus\mathbf{C}|\,\middle|\,\boldsymbol{\mathcal{E}}\right]=\sum_{x\in L}\mathbf{Pr}\left[\mathbf{v}_k(x)\neq\mathbf{v}\,\middle|\,\boldsymbol{\mathcal{E}}\right]\leq|L|\cdot 2^{k-1}\cdot\frac{d/(2^k\log s)}{d-\|a-b\|_1}\leq\frac{|L|}{\log s}, \tag{29}$$

because there are 2^{k-1} coordinates sampled up to (but not including) depth k. The last inequality above also used the assumption that $\|a-b\|_1\leq d/2$. Then, we have

$$\mathbf{Pr}\left[|\mathbf{C}|\geq\frac{|L|}{10}\,\middle|\,\boldsymbol{\mathcal{E}}\right]\geq\mathbf{Pr}\left[|L\cap\mathbf{C}|\geq\frac{|L|}{10}\,\middle|\,\boldsymbol{\mathcal{E}}\right]=1-\mathbf{Pr}\left[|L\setminus\mathbf{C}|\geq\frac{9|L|}{10}\,\middle|\,\boldsymbol{\mathcal{E}}\right]\geq 1-\frac{10}{9\log s}, \tag{30}$$

where the last inequality follows by applying Markov's inequality to (29). Hence, by an application of triangle inequality, as well as the fact that $|\mathbf{C}| \geq 1$ (since it always contains a),

$$\|a - \mathbf{c}\|_1 \leq \frac{1}{|\mathbf{C}|} \sum_{x \in \mathbf{C}} \|a - x\|_1 \leq \frac{d\alpha}{2^{k+1}} + \frac{1}{|\mathbf{C}|} \sum_{x \in \mathbf{C}} \mathbf{1}\left\{\|a - x\|_1 \geq \frac{d\alpha}{2^{k+1}}\right\} \cdot \|a - x\|_1.$$

Thus, we have (by splitting into two cases of $|\mathbf{C}| \geq |L|/10$ and $|\mathbf{C}| < |L|/10$ and applying (30))

$$\begin{aligned}\mathbf{E}\left[\|a - \mathbf{c}\|_1 \,\middle|\, \boldsymbol{\mathcal{E}}\right] &\leq \frac{d\alpha}{2^{k+1}} + \mathbf{E}\left[\frac{1}{|\mathbf{C}|} \sum_{x \in \mathbf{C}} \mathbf{1}\left\{\|a - x\|_1 \geq \frac{d\alpha}{2^{k+1}}\right\} \|a - x\|_1 \,\middle|\, \boldsymbol{\mathcal{E}}\right] \\ &\leq \frac{d\alpha}{2^{k+1}} + \frac{10}{|L|} \cdot \mathbf{E}\left[\sum_{x \in \mathbf{C}} \mathbf{1}\left\{\|a - x\|_1 \geq \frac{d\alpha}{2^{k+1}}\right\} \|a - x\|_1 \,\middle|\, \boldsymbol{\mathcal{E}}\right] + \frac{10}{9 \log s} \cdot \frac{10 d \log s}{2^k},\end{aligned}$$

where the final term used the fact that $\|a - \mathbf{c}\|_1$ is always at most $10d \log s / 2^k$. Combining (28) and the inequality above, we have

$$\frac{d}{2^{k+1}}\left(\frac{\alpha}{10} - \frac{20}{9}\right)|L| \leq \mathbf{E}\left[\sum_{x \in \mathbf{C}} \mathbf{1}\left\{\|a - x\|_1 \geq \frac{d\alpha}{2^{k+1}}\right\} \|a - x\|_1 \,\middle|\, \boldsymbol{\mathcal{E}}\right], \tag{31}$$

and we next upperbound the right-hand side above in terms of the size of H. In particular, let H' be the set of points $x \in H$ with $\|a - x\|_1 \geq d\alpha/2^{k+1}$. Then

$$\mathbf{E}\left[\sum_{x \in \mathbf{C}} \mathbf{1}\left\{\|a - x\|_1 \geq \frac{d\alpha}{2^{k+1}}\right\} \|a - x\|_1 \,\middle|\, \boldsymbol{\mathcal{E}}\right] \leq \sum_{x \in H'} \mathbf{Pr}\left[x \in \mathbf{C} \,\middle|\, \boldsymbol{\mathcal{E}}\right] \cdot \left(\frac{10 d \log s}{2^k}\right). \tag{32}$$

We consider two cases: $i = k$ and $i < k$. For the easier case when $i = k$, in order for $x \in \mathbf{C}$ to occur conditioned on $\boldsymbol{\mathcal{E}}$, we have that all 2^{k-1} coordinates sampled before depth k avoid separating x and a conditioning on not separating a and b. The conditional probability is at most

$$\frac{d - \|a - x\|_1}{d - \|a - b\|_1} = 1 - \frac{\|a - x\|_1 - \|a - b\|_1}{d - \|a - b\|_1} \leq 1 - \frac{\|a - x\|_1 - \|a - b\|_1}{d}.$$

Therefore in this case we have

$$\mathbf{Pr}\left[x \in \mathbf{C} \,\middle|\, \boldsymbol{\mathcal{E}}\right] \leq \left(1 - \frac{\|a - x\|_1 - \|a - b\|_1}{d}\right)^{2^{k-1}}.$$

Notice that by definition of $h_{a,b}$, and the fact that $k \leq h_{a,b}$, we have

$$\|x - a\|_1 - \|a - b\|_1 \geq \frac{d}{2^{k+1}}(\alpha - 2) \tag{33}$$

which implies that $\mathbf{Pr}[x \in \mathbf{C} \,|\, \boldsymbol{\mathcal{E}}]$ is at most $\exp(-\alpha/8)$ using $\alpha \geq 300$.

Next we deal with the case when $i < k$. In order for $x \in \mathbf{C}$ to occur conditioned on $\boldsymbol{\mathcal{E}}$, it needs to be the case that all coordinates sampled before (i, ℓ), of which there are $2^{i-1} + \ell - 1$ many (or none if $(i, \ell) = (0, 1)$), avoid separating x and a conditioning on not separating a and b; the (i, ℓ)-th sample is j (which better not separate x and a; otherwise the probability is trivially 0); and the remaining

$2^{k-1} - 2^{i-1} - \ell$ (or $2^{k-1} - 1$ if $(i, \ell) = (0, 1)$) coordinates do not separate x and a (but there will be conditioning on not separating a and b). Hence,

$$\begin{aligned}
\mathbf{Pr}\left[x \in \mathbf{C} \,\middle|\, \boldsymbol{\mathcal{E}}\right] &\leq \left(1 - \frac{\|x-a\|_1 - \|a-b\|_1}{d}\right)^{2^{i-1}+\ell-1} \left(1 - \frac{\|x-a\|_1}{d}\right)^{2^{k-1}-2^{i-1}-\ell} \\
&\leq \left(1 - \frac{\|x-a\|_1 - \|a-b\|_1}{d}\right)^{2^{k-1}-1} \\
&\leq \left(1 - \frac{\alpha - 2}{2^{k+1}}\right)^{2^{k-1}-1}, \qquad (34)
\end{aligned}$$

which is at most $\exp(-\alpha/8)$ using $k \geq 2$ and $\alpha \geq 300$. Hence, we can combine (31) and (32) to get

$$\frac{d}{2^{k+1}}\left(\frac{\alpha}{10} - \frac{20}{9}\right)|L| \leq |H| \cdot \exp\left(-\frac{\alpha}{8}\right) \cdot \frac{10d \log s}{2^k}.$$

Re-arranging the inequality, the lemma follows using $\alpha \geq 300$. ■

The next lemma helps bound the number of large $Q_{k,j}$'s.

Lemma 4.2. *Fix any $j \in [d]$ with $a_j \neq b_j$. For any $\alpha \geq 20 \log\log s$, the set*

$$G_j(\alpha) = \left\{k \in \{0, \ldots, h_{a,b}\} : Q_{k,j} \geq \frac{d}{2^k} \cdot \alpha\right\} \quad \textit{satisfies} \quad \left|G_j(\alpha)\right| \leq O\left(\left\lceil \frac{16 \log(2s)}{\alpha} \right\rceil \cdot \log\log s\right).$$

Proof: Assume for a contradiction that

$$\left|G_j(\alpha)\right| \geq \beta \cdot \left\lceil \log_2(10 \log^2 s)\right\rceil + 2, \quad \text{where} \quad \beta \stackrel{\text{def}}{=} \left\lceil \frac{16 \log(2s)}{\alpha} \right\rceil.$$

Then there must be $k_1, \ldots, k_\beta \in \{2, \ldots, h_{a,b}\}$ with $k_1 > k_2 > \cdots > k_\beta$ and every $t \in [\beta - 1]$ satisfies

$$k_t - k_{t+1} \geq \left\lceil \log_2(10 \log^2 s)\right\rceil.$$

This implies that every $t \in [\beta - 1]$ satisfies

$$\frac{10d \log s}{2^{k_t}} \leq \frac{d}{2^{k_{t+1}} \log s}.$$

If we consider for each $t \in [\beta]$, the following two multi-sets

$$L_t = \left\{x \in A_j \cup B_j : \|a - x\|_1 \leq \frac{d}{2^{k_t} \log s}\right\} \quad \text{and} \quad H_t = \left\{x \in A_j \cup B_j : \|a - x\|_1 \leq \frac{10d \log s}{2^{k_t}}\right\},$$

they satisfy

$$L_1 \subseteq H_1 \subseteq L_2 \subseteq H_2 \subseteq \cdots \subseteq H_{\beta-1} \subseteq L_\beta \subseteq H_\beta, \qquad (35)$$

but then invoking Lemma 4.1 (and using $20 \log\log s \geq 300$), we have that every $t \in [\beta]$ satisfy

$$|H_t| \geq \frac{|L_t| \exp(\alpha/8)}{\log s}.$$

Using $|L_1| \geq 1$ (since it contains a) and (35), we have

$$|H_\beta| \geq \frac{\exp(\alpha\beta/8)}{(\log s)^\beta} > 2s,$$

using $\exp(\alpha\beta/8) \geq (2s)^2$, and $(\log s)^\beta \leq 2s$, by our settings of β and α. This is a contradiction, as we have $H_\beta \subseteq A \cup B$ and thus, $|H_\beta| \leq 2s$. ■

Finally we finish the proof of (27). Let $\alpha_0 = 20 \log\log s$. Use Lemma 4.2 we have

$$\begin{aligned}\sum_{k=i_0-1}^{h_{a,b}} Q_{k,j} \cdot 2^k &\leq (h_{a,b} - i_0 + 2) \cdot \alpha_0 d + \sum_{\kappa=0}^{\lceil \log_2(10 \log s) \rceil} \left|G_j(\alpha_0 2^\kappa)\right| \cdot \alpha_0 2^{\kappa+1} \cdot d \\ &\leq O(d) \cdot \Big((h_{a,b} - i_0) \cdot \log\log s + \log s \cdot (\log\log s)^3\Big),\end{aligned}$$

where the upper limit of $\kappa \leq \lceil \log_2(10 \log s) \rceil$ comes from the fact that $G_j(10 \log s)$ is empty because $\|a - \widetilde{\mathbf{c}}(k, a)\|_1$ is always at most $10d \log s / 2^k$ by definition. This completes Lemma 3.5.

5 Two-Round Linear Sketch

In this section and the following, we leverage our improved analysis to design linear sketches for EMD over $\{0,1\}^d$; the extension to EMD over $(\mathbb{R}^d, \|\cdot\|_p)$ for $p \in [1,2]$ follows from a standard application of metric embeddings (see Appendix E). Specifically, we first demonstrate how a $\tilde{O}(\log s)$ approximation can be obtained using $\mathrm{polylog}(s, d)$ bits of space in a *two-round* linear sketching model (we give a formal description shortly). In the subsequent section, we implement the linear sketch in a single round, at the cost of a small additive error. In cases where sets A, B do not substantially overlap, the one-round protocol yields the same multiplicative guarantee. As is relatively standard in the sketching literature, the two-round and one-round linear sketch may be used for two-round and one-round communication settings, as well as two-pass and one-pass streaming algorithms (see Appendix F for a more thorough explanation of the model).

Linear Sketching for EMD over $\{0,1\}^d$. For some $m \in \mathbb{N}$, we encode the inputs of a computational problem as a vector f in $\mathbb{R}^m$. For that encoding and a parameter $k \in \mathbb{N}$, a *linear sketch* is a distribution over $k \times m$ matrices $\mathbf{S}$, where ideally $k \ll m$, accompanied by an algorithm $\mathrm{Alg}_{\mathbf{S}}$, which receives an input $\mathbf{S}f \in \mathbb{R}^k$ and outputs an answer to the computational problem. The algorithm maintains the vector $\mathbf{S}x$ in k words of space and utilizes $\mathrm{Alg}_{\mathbf{S}}$ to produce an output. In the public coin model, storing the random matrix $\mathbf{S}$ is not counted against the space complexity of the algorithm ($\mathbf{S}$ can often be stored in small space, such as for streaming, see Corollary F.1). In order to define linear sketching for EMD over size-s subsets of $\{0,1\}^d$, we must specify the encoding in $\mathbb{R}^m$, as well as the space complexity k, and the distribution over $k \times m$ matrices $\mathbf{S}$ and decoding algorithms.

A pair of multi-sets $A, B \subset \{0,1\}^d$ is encoded as a vector $f_{A,B} \in \mathbb{R}^{2 \cdot 2^d}$, where the first 2^d coordinates represent the indicator vector of A in $\{0,1\}^d$, and the second 2^d coordinates represent the indicator vector of B in $\{0,1\}^d$. More precisely, for $i \in \{0,1\}^d$, the i-th coordinate (in standard binary

encoding) of $f_{A,B}$ is the number of points in A at i; the (2^d+i)-th coordinate of $f_{A,B}$ is the number of points in B at i.

The two-round linear sketch is defined by a linear sketch which produces an intermediate output for the first round, and another linear sketch (parametrized by the output of the first round) which produces the final output. Specifically, we have distribution $\mathcal{D}^{(1)}$ supported on $k \times 2 \cdot 2^d$ matrices, as well a decoding algorithm for a vector $\boldsymbol{y}_1 = \mathbf{S}_1 f_{A,B} \in \mathbb{R}^k$. In one-round sketching, the output is produced by the decoding algorithm. For two-round round linear sketching, we consider another distribution $\mathcal{D}^{(2)}$, parametrized by the decoding of $\boldsymbol{y}_1$, as well as $\mathbf{S}_1$, supported on $k \times 2 \cdot 2^d$ matrices $\mathbf{S}_2$, as well as a decoding algorithm for a vector $\boldsymbol{y}_2 = \mathbf{S}_2 f_{A,B} \in \mathbb{R}^k$ which produces the output.

5.1 The Two-Round Linear Sketching Algorithm

In this section, we give a two-round linear sketching algorithm for approximating the Earth Movers distance. Recall that for a linear sketch, the multisets $A, B \subset \{0,1\}^d$ are *implicitly* encoded as a vector $f_{A,B} \in \mathbb{R}^{2^{d+1}}$ which indicates the members and frequencies of the points in A, B. Specifically, we prove the following theorem.

Theorem 6. *For $d, s \in \mathbb{N}$, there exists a 2-round linear sketching algorithm such that given multi-sets $A, B \subset \{0,1\}^d$ with $|A| = |B| = s$, computes an approximation $\widehat{\boldsymbol{\mathcal{I}}}$ to $\mathrm{EMD}(A,B)$ with*

$$\mathrm{EMD}(A,B) \leq \widehat{\boldsymbol{\mathcal{I}}} \leq \tilde{O}(\log s)\,\mathrm{EMD}(A,B)$$

with probability at least $3/4$. Moreover, the space of the linear sketch is $\mathrm{polylog}(s,d)$ bits.

The protocol will proceed by sampling a compressed quadtree $\mathbf{T}$ of depth $h = \log_2(2d)$, and for each depth $i \in \{0, \ldots, h-1\}$, using ℓ_1-sampling and ℓ_1-sketching to estimate

$$\boldsymbol{\mathcal{I}}_i \stackrel{\text{def}}{=} \sum_{v \in \mathcal{V}_i} \big||A_v| - |B_v|\big| \cdot \boldsymbol{q}_{\pi(v),v},$$

up to a constant factor. We recall that for a node v in the tree, the multi-sets

$$A_v = \{a \in A : \mathbf{v}_i(a) = v\} \qquad \text{and} \qquad B_v = \{b \in B : \mathbf{v}_i(b) = v\}$$

are the points from A and B which are mapped to node v in the compressed quadtree $\mathbf{T}$. The point

$$\mathbf{c}_v = \frac{1}{|A_v \cup B_v|} \sum_{x \in A_v \cup B_v} x,$$

is the center of mass among all points which map to node v. We have that

$$\boldsymbol{q}_{u,v} = \begin{cases} \frac{d}{2^i} & \|\mathbf{c}_u - \mathbf{c}_v\|_1 \leq \frac{d}{2^i} \\ \frac{30d\log s}{2^i} & \|\mathbf{c}_u - \mathbf{c}_v\|_1 \geq \frac{30d\log s}{2^i} \\ \|\mathbf{c}_u - \mathbf{c}_v\|_1 & \text{o.w.} \end{cases}. \tag{36}$$

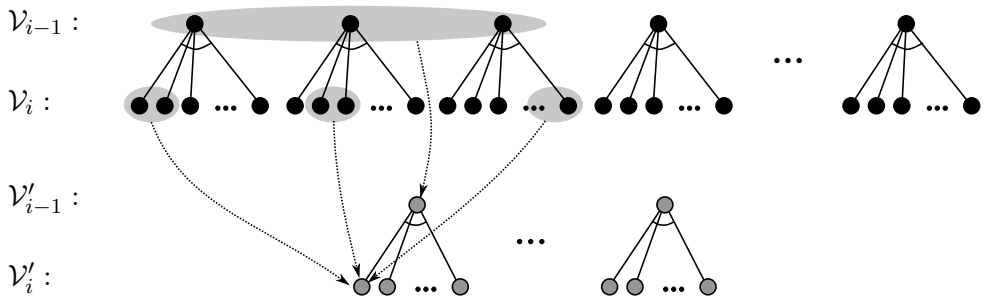

Figure 3: An example of the universe reduction from the bipartite graph defined on $(\mathcal{V}_{i-1}, \mathcal{V}_i)$ to the bipartite graph defined on $(\mathcal{V}'_{i-1}, \mathcal{V}'_i)$, as well as the corresponding mapping from $\boldsymbol{h}_{i-1}$ and $\boldsymbol{h}_i$. The shaded regions correspond to pre-images of the node mappings.

Recall that by Remark 5,

$$\mathrm{EMD}(A, B) \leq \sum_{i=0}^{h-1} \boldsymbol{\mathcal{I}}_i \leq \tilde{O}(\log s) \cdot \mathrm{EMD}(A, B),$$

with probability at least 0.89 over the draw of $\mathbf{T}$. Hence, the protocol produces an estimate $\widehat{\boldsymbol{\mathcal{I}}}_i \in \mathbb{R}$ where $\mathcal{I}_i \leq \widehat{\boldsymbol{\mathcal{I}}}_i \leq O(1) \cdot \mathcal{I}_i$ with probability at least $1 - 1/(100h)$. Via a union bound over all h levels of the tree as well as the good draw of $\mathbf{T}$, we have that with probability at least 0.88, $\sum_{i=0}^{h-1} \widehat{\boldsymbol{\mathcal{I}}}_i$ satisfies the requirements of Theorem 6. See Figure 4 for a description of the protocol which estimates $\boldsymbol{\mathcal{I}}_i$ up to constant factor with high (constant) probability.

5.1.1 Universe Reduction

Observe that $\boldsymbol{\mathcal{I}}_i$ is a sum over $|\mathcal{V}_i| = 2^{2^i}$ vertices of $\mathbf{T}$. However, since $|A| = |B| = s$, at most $2s$ of these summands will be non-zero. Thus, we perform a standard universe reduction on the layer $(\mathcal{V}_{i-1}, \mathcal{V}_i)$ as follows. Select two fully independent hash functions $\boldsymbol{h}_{i-1} : [\mathcal{V}_{i-1}] \to [s'], \boldsymbol{h}_i : [\mathcal{V}_i] \to [s']$, where $s' = O(s^3)$. We then define

$$\mathcal{V}'_{i-1} = [s'], \qquad \text{and} \qquad \mathcal{V}'_i = [s'] \times [s'],$$

and construct the bipartite graph $(\mathcal{V}'_{i-1}, \mathcal{V}'_i)$ corresponding to layers $i-1$ and i of $\mathbf{T}$ after a universe reduction. The nodes of $\mathcal{V}'_{i-1}$ represent groups of nodes in $\mathcal{V}'_{i-1}$ under the mapping $\boldsymbol{h}_{i-1}$. The nodes of $\mathcal{V}'_i$ are indexed by pairs $(y, z) \in [s'] \times [s']$, where the index y specifies the group of the parent node (under $\boldsymbol{h}_{i-1}$), and z represents group of child nodes in $\mathcal{V}_i$. The edges are added to preserve parent-child relationships in $\mathcal{V}'_{i-1}$ and $\mathcal{V}'_i$: $(x, (y, z)) \in \mathcal{V}'_{i-1} \times \mathcal{V}'_i$ is added as an edge if $x = y$ for $x, y, z \in [s']$. (See Figure 3.)

For $x \in \mathcal{V}'_{i-1}$ and $(y, z) \in \mathcal{V}'_i$, we can naturally define

$$A_x = \{a \in A : \boldsymbol{h}_{i-1}(\mathbf{v}_{i-1}(a)) = x\} \quad A_{(y,z)} = \{a \in A : \boldsymbol{h}_i(\mathbf{v}_i(a)) = z, \boldsymbol{h}_{i-1}(\mathbf{v}_{i-1}(a)) = y\}$$

We similarly define B_u, B_v for $u \in \mathcal{V}'_{i-1}, v \in \mathcal{V}'_i$. We can also define centers $\mathbf{c}'_v, \mathbf{c}'_u$ of the sets $A_u \cup B_u, A_v \cup B_v$ where $u \in \mathcal{V}'_{i-1}, v \in \mathcal{V}'_i$, and define $\boldsymbol{q}'_{\pi(v),v}$ identically to $\boldsymbol{q}'_{\pi(v),v}$ but using the

centers $\mathbf{c}'_v$. Altogether, we can set

$$\boldsymbol{\mathcal{I}}'_i \stackrel{\text{def}}{=} \sum_{v \in \mathcal{V}'_i} \big||A_v| - |B_v|\big| \cdot \boldsymbol{q}'_{\pi(v),v}$$

where $\pi(v) \in \mathcal{V}'_{i-1}$ is the parent of v in the new tree. Notice that since both $\mathcal{V}_i, \mathcal{V}_{i-1}$ have at most $2s$ non-empty nodes each, with probability $1 - 1/s$ we have that $\boldsymbol{h}_{i-1}$ and $\boldsymbol{h}_i$ are injective into $[s']$ and $[s']^2$ respectively on the non-empty nodes (i.e., the non-empty nodes perfectly hash). Since $\boldsymbol{\mathcal{I}}_i$ is just a sum over edges, if the non-empty edges perfectly hash, the construction of $\boldsymbol{\mathcal{I}}'_i$ just amounts to a renaming of the non-zero edges.

Proposition 5.1. *With probability $1 - 1/s$ over the choice of $\boldsymbol{h}_{i-1}, \boldsymbol{h}_i$, we have $\boldsymbol{\mathcal{I}}_i = \boldsymbol{\mathcal{I}}'_i$. Moreover, the hash functions $\boldsymbol{h}_{i-1}, \boldsymbol{h}_i$ need only be 2-wise independent.*

Proof: This simply follows from the fact that s' is at least $8s^3$, and there are at most $2s$ nodes $v \in \mathcal{V}_{i-1}$ where $A_v \cup B_v \neq \emptyset$. Hence, the probability that two non-empty nodes collide in $\boldsymbol{h}_{i-1}$ is at most $1/(8s^3)$ (where we use 2-wise independence), and we may union bound over at most $4s^2$ pairs of non-empty nodes. The same argument can be made that $\boldsymbol{h}_i$ perfectly hashes all non-empty nodes in $\mathcal{V}_i$ with probability $1 - 1/(2s)$. Conditioned on this, there is a bijection between non-zero terms in $\boldsymbol{\mathcal{I}}_i$ and $\boldsymbol{\mathcal{I}}'_i$, where terms mapped to each other are equal. ■

Proposition 5.1 demonstrates that it will be sufficient to estimate $\boldsymbol{\mathcal{I}}'_i$. Thus, we condition on the success of this universe reduction step now. Since conditioned on this step the non-zero edges $(\mathcal{V}_{i-1}, \mathcal{V}_i)$ are in bijective correspondence with the non-zero edges of $(\mathcal{V}'_{i-1}, \mathcal{V}'_i)$, and moreover since under this equivelance the points contained in A_v, B_v and the centers $\mathbf{c}_v$ are equivalant, in the following we will abuse notation and drop the prime notation in $\boldsymbol{\mathcal{I}}'_i, \mathcal{V}'_i, \mathcal{V}'_{i-1}$, and simply write $\boldsymbol{\mathcal{I}}_i, \mathcal{V}_i, \mathcal{V}_{i-1}$, with the understanding that the universe reduction step has already been carried out.

5.1.2 Sketching Tools.

We will need the following two ingredients for the protocol and the proof.

Theorem 7 (ℓ_1-sketch [Ind06a])**.** *For $m \in \mathbb{N}$ and $\delta \in (0, 1/2)$, let $t = O(\log(1/\delta))$. There exists a distribution $\mathcal{S}^1_k(m, t)$ supported on $t \times m$ matrices $\mathbf{C}$, as well as a recovery algorithm Alg^1 which receives a vector $y \in \mathbb{R}^t$ and outputs a real number. For any fixed $x \in \mathbb{R}^m$,*

$$\|x\|_1 \leq \mathrm{Alg}^1(\mathbf{C}x) \leq 2\|x\|_1,$$

with probability at least $1 - \delta$ over the draw of $\mathbf{C} \sim \mathcal{S}^1_k(m, t)$.[19]

Theorem 8 (Perfect ℓ_1-sampling [JW18])**.** *For $m \in \mathbb{N}$, let $c > 1$ be an arbitrarily large constant and $t = O(\log^2 m)$. There exists a distribution $\mathcal{S}^1_a(m, t)$ supported on pairs $(\mathbf{S}, \mathrm{Alg}^1_{\mathbf{S}})$ where $\mathbf{S}$ is an $t \times m$ matrix and $\mathrm{Alg}^1_{\mathbf{S}}$ is an algorithm which receives as input a vector $y \in \mathbb{R}^t$ and outputs a failure*

[19] The notation for the distribution $\mathcal{S}^1_k$, is for ℓ_1-sketching; the 1 in the superscript for the ℓ_1 aspect, and k in the subscript is for "sketching". Later, we will see analogous $\mathcal{S}^\infty_k$ for "ℓ_∞-sketching" (i.e., Count-Sketch), and $\mathcal{S}^1_a$ for ℓ_1-sampling.

symbol $\perp$ *with probability at most* $1/3$*, otherwise it returns an index* $j \in [m]$*. For any* $x \in \mathbb{R}^m$ *and* $j \in [m]$*,*

$$\left| \mathop{\mathbf{Pr}}_{(\mathbf{S},\mathrm{Alg}^1_{\mathbf{S}})\sim\mathcal{S}^1_a(m,t)} \left[\mathrm{Alg}^1_{\mathbf{S}}(\mathbf{S}x) = j \mid \mathrm{Alg}^1_{\mathbf{S}}(\mathbf{S}x) \neq \perp\right] - \frac{|x_j|}{\|x\|_1} \right| \leq \frac{1}{m^c}.$$

5.1.3 Description and Analysis of Two-Round Sketch

We give a description of the two-round protocol, and the precise formulation is given in Figure 4. As we explain the protocol we specify some claims, which are then combined to prove Theorem 6.

Round 0. We begin by mapping the vector $f_{A,B} \in \mathbb{R}^{2^{d+1}}$ to a vector $f^i \in \mathbb{R}^{|\mathcal{V}_i|\times\{A,B\}}$ indexed by vertices $v \in \mathcal{V}_i$ and $\{A, B\}$ via

$$f^i_{v,A} = \sum_{\substack{\vec{x}\in\{0,1\}^d \\ v_i(\vec{x})=v}} (f_{A,B})_x = |A_v| \quad \text{and} \quad f^i_{v,B} = \sum_{\substack{\vec{x}\in\{0,1\}^d \\ v_i(\vec{x})=v}} (f_{A,B})_{x+2^d} = |B_v|$$

where $\vec{x} \in \{0,1\}^d$ and $x \in [2^d]$ is the index in $[2^d]$ that $\vec{x}$ encodes. In other words, $f^i_{v,A}$ and $f^i_{v,B}$ hold the frequency counts of points in A and B, respectively, at vertex $v \in \mathcal{V}_i$, which is a linear mapping of the input $f_{A,B}$. Recall we assume the universe reduction has already been performed, so $|\mathcal{V}_i| = O(s^6)$. At this point, we initialize sketches to perform ℓ_1-sampling and ℓ_1-sketching.

Round 1. At the end of round 1, the estimate $\widehat{\boldsymbol{\beta}} \in \mathbb{R}$ is the result of an ℓ_1-sketch on the vector F^i. In other words, we will have that $\widehat{\boldsymbol{\beta}}$ is a constant factor approximation to $\|F^i\|_1$. Furthermore, the indices $v_1, \ldots, v_k \in \mathcal{V}_i \cup \{\perp\}$ represent ℓ_1-samples from F^i; while the ℓ_1-sample does not fail (i.e., $v_t \neq \perp$), the sample $v_t \in \mathcal{V}_i$ is meant to be distributed proportional to $F^i_{v_t}$. Specifically, we have the following two claims:

Claim 5.2 (Round 1 Claim A)**.** *Let* $\boldsymbol{\mathcal{E}}_1$ *be the event, defined over the randomness of the sample* $\mathbf{C} \sim \mathcal{S}^1_k((s')^3, O(1))$ *in Line 3 of Round 0 that in Line 2 of Round 1,*

$$\sum_{v\in\mathcal{V}_i} \big||A_v| - |B_v|\big| \leq \widehat{\boldsymbol{\beta}} \leq 2\sum_{v\in\mathcal{V}_i} \big||A_v| - |B_v|\big|. \tag{37}$$

Then, $\boldsymbol{\mathcal{E}}_1$ *occurs with probability at least* 0.99*.*

Proof: This is a consequence of Theorem 7 with $m = (s')^2$, $\delta = .01$, applied to the vector F^i which has ℓ_1 norm $\|F^i\|_1 = \sum_{v\in\mathcal{V}_i} \big||A_v| - |B_v|\big|$ ■

Claim 5.3 (Round 1 Claim B)**.** *The random variables* $v_1, \ldots, v_k$ *defined in Line 2 are independent and identically distributed. We have that* $v_t = \perp$ *with probability at most* $1/3$ *for each* t*. Otherwise, for any* $v \in \mathcal{V}_i$ *we have*

$$\left| \mathbf{Pr}\left[v_t = v\right] - \frac{\big||A_v| - |B_v|\big|}{\sum_{v'\in\mathcal{V}_i} \big||A_{v'}| - |B_{v'}|\big|} \right| \leq \frac{1}{s^{10}}.$$

Proof: We first note that independence follows from the fact that the random variables $v_1, \ldots, v_k$ depend on the independent samples $(\mathbf{S}_t, \mathrm{Alg}^1_{\mathbf{S}_t}) \sim \mathcal{S}^1_a((s')^2, O(\log^2 s))$ for $t \in [k]$ of Line 2 of Round 0. The distributional bound then follows from Theorem 8 with $c = 10$, applied to the input vector F^i.

■

Round 2. The second round is meant to communicate, for each $t \in [k]$, the information necessary to estimate $\boldsymbol{q}_{\pi(v_t),v_t}$. In order to do this, the players must jointly approximate $\|c_{\pi(v_t)} - c_{v_t}\|_1$. For each $t \in [k]$, n_t is the number of points in A and B which lie in $\pi(v_t)$, and notice that, once both players know v_t, n_t can be computed exactly by communicating $|A_{\pi(v_t)}|$ and $|B_{\pi(v_t)}|$ with $O(\log s)$ bits. The centers of mass, c_{v_t} and $c_{\pi(v_t)}$ are given by the vectors

$$c_{v_t} = \frac{1}{f^i_{v_t,A} + f^i_{v_t,B}} \sum_{\substack{\vec{x}\in\{0,1\}^d \\ v_i(\vec{x})=v_t}} \vec{x}\left((f_{A,B})_x + (f_{A,B})_{x+2^d}\right)$$

$$c_{\pi(v_t)} = \frac{1}{n_t} \sum_{\substack{\vec{x}\in\{0,1\}^d \\ v_{i-1}(\vec{x})=\pi(v_t)}} \vec{x}\left((f_{A,B})_x + (f_{A,B})_{x+2^d}\right).$$

Therefore, $\boldsymbol{\chi}^i_t$ is an ℓ_1-sketch of $(f^i_{v_t,A} + f^i_{v_T,B})c_{v_t}$ and $\boldsymbol{\chi}^{i-1}_t$ an ℓ_1-sketch of $n_t c_{\pi(v_t)}$ such that $\|c_{\pi(v_t)} - c_{v_t}\|_1$ can be approximated, and $\widehat{\boldsymbol{q}}_t$ becomes an approximation of $\boldsymbol{q}_{\pi(v_t),v_t}$. This last point is captured by the following claim.

Claim 5.4 (Round 2 Claim)**.** *Let $v_t \in \mathcal{V}_i$ be any sample returned by Line 2 in Round 1, and suppose $v_t \neq \perp$. Let $\boldsymbol{\mathcal{E}}_{2,t}$ be the event, defined over the randomness of the sample $\mathbf{C}_t \sim \mathcal{S}^1_k(d, O(\log\log s))$ in Line 3 of Round 0 that in Line 3 of Round 2,*

$$\|\mathbf{c}_{v^*} - \mathbf{c}_{u^*}\|_1 \leq \mathrm{Alg}^1\left(\frac{\boldsymbol{\chi}^{i-1}_t}{\boldsymbol{n}_t} - \frac{\boldsymbol{\chi}^i_t}{f^i_{v_t,a} + f^i_{v_t,b}}\right) \leq 2\|\mathbf{c}_{v^*} - \mathbf{c}_{u^*}\|_1.$$

Then, $\boldsymbol{\mathcal{E}}_{2,t}$ occurs with probability at least $1 - 1/(100k)$.

Proof: We apply Theorem 7 with $m = d$, $t = O(\log\log s)$, and notice by definition

$$\frac{\boldsymbol{\chi}^{i-1}_t}{n_t} = \frac{\sum_{\substack{\vec{x}\in\{0,1\}^d \\ v_{i-1}(x)=\pi(v_t)}} (\mathbf{C}_t\vec{x}) \cdot ((f_{A,B})_x + (f_{A,B})_{x+2^d})}{|A_{\pi(v_t)}| + |B_{\pi(v_t)}|} = \frac{\sum_{x\in A_{\pi(v_t)}\cup B_{\pi(v_t)}} \mathbf{C}_t x}{|A_{\pi(v_t)}| + |B_{\pi(v_t)}|} = \mathbf{C}_t \cdot c_{\pi(v_t)}$$

similarly

$$\frac{\boldsymbol{\chi}^i_t}{f^i_{v_t,a} + f^i_{v_t,b}} = \frac{\sum_{\substack{\vec{x}\in\{0,1\}^d \\ v_i(x)=v_t}} (\mathbf{C}_t\vec{x}) \cdot ((f_{A,B})_x + (f_{A,B})_{x+2^d})}{|A_{v_t}| + |B_{v_t}|} = \frac{\sum_{x\in A_{v_t}\cup B_{v_t}} \mathbf{C}_t x}{|A_{v_t}| + |B_{v_t}|} = \mathbf{C}_t \cdot c_{v_t}$$

thus, by linearity

$$\frac{\boldsymbol{\chi}^{i-1}_t}{n_t} - \frac{\boldsymbol{\chi}^i_t}{f^i_{v_t,a} + f^i_{v_t,b}} = \mathbf{C}_t(c_{v_t} - c_{\pi(v_t)})$$

where $\mathbf{C}_t \sim \mathcal{S}_k^1(d, O(\log\log s))$, and noting that $k = O(\log s)$ so the failure probability of Theorem 7 is $1/\mathrm{poly}(k)$. ■

Two-round Sketch $(f_{A,B}, T, i)$

Input: Vector $f_{A,B} \in \mathbb{R}^{2^{d+1}}$ encoding two multi-sets $A, B \subset \{0,1\}^d$, a compressed quadtree T, and index $i \in \{0, \ldots, h-1\}$.
Output: A real number $\widehat{\boldsymbol{\mathcal{I}}}_i$.

- Round 0: Perform the universe reduction step (Proposition 5.1) so that $|\mathcal{V}_{i-1}| = s', |\mathcal{V}_i| = (s')^2$, where $s' = O(s^3)$. Set $k = O(\log^2 s)$.
  1. Define the linear functions $f^i \in \mathbb{R}^{|\mathcal{V}_i| \times \{A,B\}}$ and $F^i \in \mathbb{R}^{|\mathcal{V}_i|}$ of $f_{A,B}$ via
  $$f^i_{v,A} = \sum_{\substack{\vec{x} \in \{0,1\}^d \\ v_i(\vec{x}) = v}} (f_{A,B})_x = |A_v| \quad \text{and} \quad f^i_{v,B} = \sum_{\substack{\vec{x} \in \{0,1\}^d \\ v_i(\vec{x}) = v}} (f_{A,B})_{x+2^d} = |B_v|$$
  $$F^i_v = f^i_{v,A} - f^i_{v,B} = |A_v| - |B_v|$$
  2. For each $t \in [k]$, initialize linear sketch $(\mathbf{S}_t, \mathrm{Alg}^1_{\mathbf{S}_t}) \sim \mathcal{S}^1_a((s')^2, O(\log^2 s))$.
  3. For each $t \in [k]$, initialize linear sketch $\mathbf{C}_t \sim \mathcal{S}^1_k(d, O(\log\log s))$, and initialize $\mathbf{C} \sim \mathcal{S}^1_k((s')^2, O(1))$.
- Round 1:
  1. Construct the linear sketch $\boldsymbol{\beta} \stackrel{\text{def}}{=} \mathbf{C}F^i$, and for each $t \in [k]$, construct the linear sketches $\boldsymbol{\alpha}_t \stackrel{\text{def}}{=} \mathbf{S}_t F^i$.
  2. Generate k samples $v_1, \ldots, v_k \in \mathcal{V}_i \cup \{\perp\}$ obtained from
  $$v_t \stackrel{\text{def}}{=} \mathrm{Alg}^1_{\mathbf{S}_t}(\boldsymbol{\alpha}_t), \qquad \text{and let} \qquad \widehat{\boldsymbol{\beta}} \stackrel{\text{def}}{=} \mathrm{Alg}^1(\boldsymbol{\beta}).$$
- Round 2: Let $L \subset [k]$ be the set of indices such that $v_t \neq \perp$. For each $t \in L$:
  1. Compute the value $n_t \stackrel{\text{def}}{=} \sum_{\substack{v \in \mathcal{V}_i \\ \pi(v) = \pi(v_t)}} f^i_{v,A} + f^i_{v,B} = |A_{\pi(v_t)}| + |B_{\pi(v_t)}|$.
  2. Compute the vectors

$$\boldsymbol{\chi}^i_t \stackrel{\text{def}}{=} \sum_{\substack{\vec{x} \in \{0,1\}^d \\ v_i(\vec{x}) = v_t}} (\mathbf{C}_t \vec{x}) \cdot ((f_{A,B})_x + (f_{A,B})_{x+2^d}), \qquad \boldsymbol{\chi}^{i-1}_t \stackrel{\text{def}}{=} \sum_{\substack{\vec{x} \in \{0,1\}^d \\ v_{i-1}(\vec{x}) = \pi(v_t)}} (\mathbf{C}_t \vec{x}) \cdot ((f_{A,B})_x + (f_{A,B})_{x+2^d}).$$

  3. Let
  $$\widehat{\boldsymbol{q}}_t \stackrel{\text{def}}{=} \max\left\{\min\left\{\mathrm{Alg}^1\left(\frac{\boldsymbol{\chi}^{i-1}_t}{n_t} - \frac{\boldsymbol{\chi}^i_t}{f^i_{v_t,A} + f^i_{v_t,B}}\right), \frac{30d\log s}{2^i}\right\}, \frac{d}{2^i}\right\},$$
  if all $\boldsymbol{n}_t$ and $f^i_{v_t,A} + f^i_{v_t,B}$ are non-zero. Output
  $$\widehat{\boldsymbol{\mathcal{I}}}_i \stackrel{\text{def}}{=} \widehat{\boldsymbol{\beta}} \cdot \frac{1}{|L|} \sum_{t \in L} \widehat{\boldsymbol{q}}_t.$$

Figure 4: The Two-round Protocol.

Proof of Theorem 6: As mentioned in the paragraph subsequent to stating Theorem 6, we show that for any compressed quadtree T and any $i \in \{0, \dots, h-1\}$, with probability at least $3/4$ over the execution of the sketching algorithm, the output $\widehat{\boldsymbol{\mathcal{I}}}_i$ satisfies $\boldsymbol{\mathcal{I}}_i/8 \le \widehat{\boldsymbol{\mathcal{I}}}_i \le 8\boldsymbol{\mathcal{I}}_i$. Repeating the protocol for $O(\log\log d)$ iterations and outputting the median (in order to decrease the error probability), and repeating it for all $i \in \{0, \dots, h-1\}$ obtains the desired theorem.

For the rest of the proof, consider a fixed $i \in \{0, \dots, h-1\}$ and T. We consider the distribution ν supported on $\mathcal{V}_i$ where

$$\mathop{\mathbf{Pr}}_{\mathbf{v}\sim\nu}[\mathbf{v} = v^*] \propto \big||A_{v^*}| - |B_{v^*}|\big|, \qquad \text{and} \qquad Z \stackrel{\text{def}}{=} \sum_{v\in\mathcal{V}_i} \big||A_v| - |B_v|\big|$$

is the normalizing constant. We have

$$\boldsymbol{\mathcal{I}}_i = Z \cdot \mathop{\mathbf{E}}_{\mathbf{v}\sim\nu}\left[\boldsymbol{q}_{\mathbf{v},\pi(\mathbf{v})}\right] \ge Z \cdot \frac{d}{2^i}.$$

To complete the proof, we show the following:

1. The random variable $\widehat{\boldsymbol{\beta}}$ satisfies $Z \le \widehat{\boldsymbol{\beta}} \le 2Z$, with probability at least 0.99 over the randomness in $\mathbf{C} \sim \mathcal{S}^1_k((s')^3, O(1))$. This follows immediately from Claim 5.2.

2. For every $t \in L$, consider the sample $v_t \in \mathcal{V}_i$ in Line 2 of Round 1. Then, with probability at least $1 - 1/(100k)$ over the draw of $\mathbf{C}_t \sim \mathcal{S}^1_k(d, O(\log\log s))$,

$$\frac{d}{2^i} \le \boldsymbol{q}_{v_j,u_j} \le \widehat{\boldsymbol{q}}_t \le 2\boldsymbol{q}_{v_j,u_j} \le \frac{60d\log s}{2^i},$$

   which follows from Claim 5.4.

3. We have $|L| > k/2$ with probability $1 - 1/s$. This follows from a Chernoff bound on the number of v_t such that $v_t = \perp$.

4. The random variables v_i for $i \in [L]$ in Line 2 are independent and identically distributed, depending on the draw of $(\mathbf{S}_t, \mathrm{Alg}^{(\ell_1\text{-sa})}_{\mathbf{S}_t}) \sim \mathcal{S}^1_a((s')^2, O(\log^2 s))$. Specifically, they are drawn from a distribution $\mathcal{D}'$ over $\mathcal{V}_i$, such that we have $d_{\mathrm{TV}}(\mathcal{D}', \nu) \le 1/s$. This follows from Claim 5.3.

From 1 and 4 above, we have that with probability at least 0.99,

$$\boldsymbol{\mathcal{I}}_i/2 \le \boldsymbol{\mathcal{I}}_i - O\left(\frac{Z\cdot d\log s}{s2^i}\right) \le \mathop{\mathbf{E}}_{\mathbf{v}\sim\mathcal{D}'}\left[\widehat{\boldsymbol{\beta}}\cdot\boldsymbol{q}_{\mathbf{v},\pi(\mathbf{v})}\right] \le \boldsymbol{\mathcal{I}}_i + O\left(\frac{Z\cdot d\log s}{s2^i}\right) \le 2\boldsymbol{\mathcal{I}}_i$$

and by 2, with probability $0.99 \cdot 0.99$ over the draws of $\mathbf{C} \sim \mathcal{S}^1_k((s')^2, O(1))$ and all $\mathbf{C}_1, \dots, \mathbf{C}_k \sim \mathcal{S}^1_k(d, O(\log\log s))$,

$$\frac{1}{4}\cdot\boldsymbol{\mathcal{I}}_i \le \mathop{\mathbf{E}}_{\mathbf{v}\sim\mathcal{D}'}\left[\widehat{\boldsymbol{\beta}}\cdot\widehat{\boldsymbol{q}}_t\right] \le 4\boldsymbol{\mathcal{I}}_i$$

Thus, the output $\widehat{\boldsymbol{q}}$ of the sketching algorithm has expectation between $\boldsymbol{\mathcal{I}}_i/4$ and $4\boldsymbol{\mathcal{I}}_i$, and by the boundedness of $\widehat{\boldsymbol{q}}_t$ and independence across $t \in [k]$,

$$\mathbf{Var}\left[\frac{\widehat{\boldsymbol{\beta}}}{k}\sum_{t=1}^{k}\widehat{\boldsymbol{q}}_t\right] = O\left(\frac{Z^2d^2\log^2 s}{k\cdot 2^{2i}}\right),$$

which is less than $c\boldsymbol{\mathcal{I}}_i^2$ for arbitrarily small constant $c > 0$, given large enough $k = O(\log^2 s)$. As a result, we apply Chebyshev's inequality to conclude that the output $\widehat{\boldsymbol{q}}$ is between $\boldsymbol{\mathcal{I}}_i/8$ and $8\boldsymbol{\mathcal{I}}_i$ with probability at least $0.99 \cdot 0.99 \cdot 0.99 > 3/4$ as needed, and we conclude the theorem. ■

6 One-Round Linear Sketch

In this section, we demonstrate how the two-round sketch of Section 5 can be compressed into a *single* round, albeit with the addition of a small additive error. Recall that a one-round linear sketch first implicitly generates a random matrix $\mathbf{S} \in \mathbb{R}^{k \times 2^{d+1}}$, and stores only the matrix vector product $\mathbf{S}f$, where $f \in \mathbb{R}^{2^{d+1}}$ is the vectorized representation of the two multi-sets $A, B \subset \{0,1\}^d$. The space used by a linear sketch is the number of bits required to store $\mathbf{S}f$. In our setting, since $\mathbf{S}$ will have small bit complexity and f is $2s$ sparse, the space complexity will be a $\log s$ factor larger than the number of rows of $\mathbf{S}$. Specfically, the goal of this section is to prove the following Theorem.

Theorem 9. *For $d, s \in \mathbb{N}$, there exists a 1-round linear sketching algorithm such that given multi-sets $A, B \subset \{0,1\}^d$ with $|A| = |B| = s$, computes an approximate $\widehat{\boldsymbol{\mathcal{I}}}$ to $\mathrm{EMD}(A, B)$ with*

$$\mathrm{EMD}(A, B) \le \widehat{\boldsymbol{\mathcal{I}}} \le \tilde{O}(\log s)\,\mathrm{EMD}(A, B) + \epsilon d s$$

with probability at least $3/4$. Moreover, the space used by the linear sketch is $O(1/\epsilon) \cdot \mathrm{polylog}(s, d)$.

Rescaling ϵ by a factor of d yields a additive ϵs approximation in $O(d/\epsilon) \cdot \mathrm{polylog}(s, d)$ space. Whenever, the size-s multi-sets $A, B \subset \{0,1\}^d$ do not intersect drastically, i.e., $|A \cap B|/|A \cup B| < 1 - \epsilon$ where $\cup, \cap$ are multi-set unions and intersections, then $\mathrm{EMD}(A, B) > \epsilon s$. This instances, also known as those having Jaccard index bounded away from 1, have been studied for EMD in low-dimensions [YO14], and in this case, we obtain the following corollary.

Corollary 6.1. *For $d, s \in \mathbb{N}$ and $\epsilon \in (0, 1)$, there exists a 1-round linear sketching algorithm such that given multi-sets $A, B \subset \{0,1\}^d$ with $|A| = |B| = s$ satisfying $|A \cap B|/|A \cup B| \le 1 - \epsilon$, computes an approximate $\widehat{\boldsymbol{\mathcal{I}}}$ to $\mathrm{EMD}(A, B)$ with*

$$\mathrm{EMD}(A, B) \le \widehat{\boldsymbol{\mathcal{I}}} \le \tilde{O}(\log s)\,\mathrm{EMD}(A, B)$$

with probability at least $3/4$. Moreover, the space used by the linear sketch is $O(d/\epsilon) \cdot \mathrm{polylog}(s, d)$.

The goal is similar to the two-round protocol. We begin by sampling a compressed quadtree $\mathbf{T}$ if depth $h = \log_2(2d)$, as well as a universe reduction step of Proposition 5.1 in Section 5, using public randomness. For each depth $i \in \{0, \ldots, h-1\}$, we estimate the quantity

$$\boldsymbol{\mathcal{I}}_i \stackrel{\text{def}}{=} \sum_{v \in \mathcal{V}_i} \big||A_v| - |B_v|\big| \cdot \boldsymbol{q}_{\pi(v),v}.$$

To estimate $\boldsymbol{\mathcal{I}}_i$, we first sample a vertex $v \sim \mathcal{V}_i$ with probability proportional to $\big||A_v| - |B_v|\big|$. In addition, we obtain an estimate $\widehat{\Delta}_i$ of $\Delta_i = \sum_{v \in \mathcal{V}_i} \big||A_v| - |B_v|\big|$. Once we have such a sample v, we approximate compute the cost $\boldsymbol{q}_{\pi(v),v}$ via a value $\widehat{\boldsymbol{q}}_{\pi(v),v}$, and output $\widehat{\Delta}_i \cdot \widehat{\boldsymbol{q}}_{\pi(v),v}$ as an approximation of $\boldsymbol{\mathcal{I}}_i$. Repeating the process $\mathrm{poly}(\log s)$ times, we will obtain our desired estimate. While in Section 5, we could use one round to produce the sample v, and another round to estimate $\widehat{\boldsymbol{q}}_{\pi(v),v}$, the main challenge here is to procedure the sample and estimate *simultaneously*. The approach is based on a new technique we call *precision sampling with meta-data*, building on [AKO10, JST11, JW18].

6.1 Exponential Order Statistics

In this section, we discuss several useful properties of the order statistics of n independent non-identically distributed exponential random variables. Let $(\boldsymbol{t}_1, \ldots, \boldsymbol{t}_n)$ be independent exponential random variables where $\boldsymbol{t}_i$ has mean $1/\lambda_i$ (equivalently, $\boldsymbol{t}_i$ has rate λ_i), abbreviated as $\text{Exp}(\lambda)$. Recall that $\boldsymbol{t}_i$ is given by the cumulative distribution function $\mathbf{Pr}[\boldsymbol{t}_i < x] = 1 - e^{-\lambda_i x}$. Our algorithm will require an analysis of the distribution of values $(\boldsymbol{t}_1, \ldots, \boldsymbol{t}_n)$, which we will now describe. We begin by noting that constant factor scalings of an exponential variable result in another exponential variable.

Fact 6.2 (Scaling of exponentials)**.** *Let $\boldsymbol{t}$ be exponentially distributed with rate λ, and let $\alpha > 0$. Then $\alpha\boldsymbol{t}$ is exponentially distributed with rate λ/α.*

Proof: The cdf of $\alpha\boldsymbol{t}$ is given by $\mathbf{Pr}[\boldsymbol{t} < x/\alpha] = 1 - e^{-\lambda x/\alpha}$, which is the cdf of an exponential with rate λ/α. ■

Definition 6.3. *Let $\boldsymbol{t} = (\boldsymbol{t}_1, \ldots, \boldsymbol{t}_n)$ be independent exponentials. For $k = 1, 2, \ldots, n$, we define the k-th anti-rank $D_{\boldsymbol{t}}(k) \in [n]$ of $(\boldsymbol{t}_1, \ldots, \boldsymbol{t}_n)$ to be the values $D_{\boldsymbol{t}}(k)$ such that $\boldsymbol{t}_{D_{\boldsymbol{t}}(1)} \leq \boldsymbol{t}_{D_{\boldsymbol{t}}(2)} \leq \cdots \leq \boldsymbol{t}_{D_{\boldsymbol{t}}(n)}$.*

Using the structure of the anti-rank vector, it has been observed [Nag06] that there is a simple form for describing the distribution of $\boldsymbol{t}_{D_{\boldsymbol{t}}(k)}$ as a function of $(\lambda_1, \ldots, \lambda_n)$ and the anti-rank vector.

Fact 6.4 ([Nag06])**.** *Let $(\boldsymbol{t}_1, \ldots, \boldsymbol{t}_n)$ be independently distributed exponentials, where each $\boldsymbol{t}_i$ has rate $\lambda_i > 0$. Then, $D_{\boldsymbol{t}}(1)$ is i with probability $\lambda_i / \sum_{j \in [n]} \lambda_j$. Furthermore, the following two sampling procedures produce the same distribution over pairs in $\mathbb{R}^2$:*

1. *We sample $(\boldsymbol{t}_1, \ldots, \boldsymbol{t}_n)$, where $\boldsymbol{t}_i \sim \text{Exp}(\lambda_i)$, and output $(\boldsymbol{t}_{D_{\boldsymbol{t}}(1)}, \boldsymbol{t}_{D_{\boldsymbol{t}}(2)} - \boldsymbol{t}_{D_{\boldsymbol{t}}(1)})$.*
2. *We sample $\boldsymbol{i}_1 \in [n]$ where $\mathbf{Pr}[\boldsymbol{i}_1 = i] = \lambda_i / \sum_{j=1}^{n} \lambda_i$, and independently sample $\boldsymbol{E}_1, \boldsymbol{E}_2 \sim \text{Exp}(1)$. We output*
$$\left(\frac{\boldsymbol{E}_1}{\sum_{j \in [n]} \lambda_j}, \frac{\boldsymbol{E}_2}{\sum_{j \in [n] \setminus \{\boldsymbol{i}_1\}} \lambda_j} \right).$$

Proof: This is a simple computation. We have that for any $r, r' \in \mathbb{R}_{\geq 0}$ and $i \in [n]$,

$$\begin{aligned}
\Pr_{\boldsymbol{t}} \left[\begin{array}{c} D_{\boldsymbol{t}}(1) = i, \\ \boldsymbol{t}_{D_{\boldsymbol{t}}(1)} \geq r, \\ \boldsymbol{t}_{D_{\boldsymbol{t}}(2)} - \boldsymbol{t}_{D_{\boldsymbol{t}}(1)} \geq r' \end{array} \right] &= \int_{y:r}^{\infty} \lambda_i \exp\left(-\lambda_i y\right) \prod_{j \in [n] \setminus \{i\}} \Pr_{\boldsymbol{t}_j \sim \text{Exp}(\lambda_j)} \left[\boldsymbol{t}_j - y \geq r' \right] dy \\
&= \lambda_i \exp\left(-r' \sum_{j \in [n] \setminus \{i\}} \lambda_j \right) \int_{y:r}^{\infty} \exp\left(-y \sum_{j=1}^{n} \lambda_j \right) dy \\
&= \frac{\lambda_i}{\sum_{j=1}^{n} \lambda_i} \cdot \exp\left(-r' \sum_{j \in [n] \setminus \{i\}} \lambda_j \right) \cdot \exp\left(-r \sum_{j=1}^{n} \lambda_i \right) \\
&= \mathbf{Pr}\left[\boldsymbol{i}_1 = i \wedge \frac{\boldsymbol{E}_1}{\sum_{j \in [n]} \lambda_j} \geq r \wedge \frac{\boldsymbol{E}_2}{\sum_{j \in [n] \setminus \{\boldsymbol{i}_1\}} \lambda_j} \geq r' \right].
\end{aligned}$$

■

6.2 Precision Sampling

Lemma 6.5. *Let $x \in \mathbb{R}^n$ be any vector, and let $\boldsymbol{z} \in \mathbb{R}^n$ be the random vector given by letting*

$$\boldsymbol{z}_i = \frac{x_i}{\boldsymbol{t}_i}, \qquad \text{where} \qquad \boldsymbol{t}_i \sim \text{Exp}(1).$$

Let $\epsilon \in (0,1)$, and suppose $\tilde{\boldsymbol{z}}$ is an adversarially corrupted vector satisfying $\|\tilde{\boldsymbol{z}} - \boldsymbol{z}\|_\infty \leq \epsilon \|x\|_1$, and define the random variable $\boldsymbol{i}^* = \arg\max_i |\tilde{\boldsymbol{z}}_i|$. Then we have for all $i \in [n]$:

$$\Pr_{\boldsymbol{t}_1,\ldots,\boldsymbol{t}_n \sim \text{Exp}(1)}[\boldsymbol{i}^* = i] = (1 \pm O(\epsilon))\frac{|x_i|}{\|x\|_1} \pm O(e^{-\frac{1}{4\epsilon}}) \tag{38}$$

Proof: Consider the random variable $\boldsymbol{t}' = (\boldsymbol{t}'_1, \ldots, \boldsymbol{t}'_n)$ where $\boldsymbol{t}'_i \sim \text{Exp}(|x_i|)$, and notice, by Fact 6.2, that $\boldsymbol{t}'_i$ is distributed as $1/|\boldsymbol{z}_i|$. Hence, $\boldsymbol{i}^*$ is set to i whenever $D_{\boldsymbol{t}'}(1) = i$ and $1/\boldsymbol{t}'_{D_{\boldsymbol{t}'}(1)} - 1/\boldsymbol{t}'_{D_{\boldsymbol{t}'}(2)} \geq 2\epsilon\|x\|_1$. Thus, by Fact 6.4,

$$\begin{aligned}
\Pr_{\substack{\boldsymbol{t}_1,\ldots,\boldsymbol{t}_n \\ \sim \text{Exp}(1)}}[\boldsymbol{i}^* = i] &\geq \Pr_{\substack{\boldsymbol{t}'=(\boldsymbol{t}'_1,\ldots,\boldsymbol{t}'_n) \\ \boldsymbol{t}'_i \sim \text{Exp}(|x_i|)}}\left[D_{\boldsymbol{t}'}(1) = i \wedge \frac{1}{\boldsymbol{t}'_{D_{\boldsymbol{t}'}(1)}} - \frac{1}{\boldsymbol{t}'_{D_{\boldsymbol{t}'}(2)}} \geq 2\epsilon\|x\|_1\right] \\
&\geq \Pr_{\boldsymbol{i}_1, \boldsymbol{E}_1, \boldsymbol{E}_2}\left[\boldsymbol{i}_1 = i \wedge \frac{1}{\boldsymbol{E}_1} - \frac{1}{\boldsymbol{E}_1 + \boldsymbol{E}_2} \geq 2\epsilon\right] \\
&= \frac{|x_i|}{\|x\|_1}\left(1 - \int_{r:0}^{1/(2\epsilon)} \exp(-r) \cdot \Pr_{\boldsymbol{E}_2}\left[\boldsymbol{E}_2 \leq \frac{2\epsilon r^2}{1-2\epsilon r}\right] dr - \exp\left(-\frac{1}{2\epsilon}\right)\right),
\end{aligned}$$

and we have

$$\int_{r:0}^{1/(2\epsilon)} \exp(-r) \Pr_{\boldsymbol{E}_2}\left[\boldsymbol{E}_2 \leq \frac{2\epsilon r^2}{1-2\epsilon r}\right] dr \leq \int_{r:0}^{1/(4\epsilon)} 4\epsilon r^2 \exp(-r) dr + \frac{1}{4\epsilon}\exp(-1/(4\epsilon)) \lesssim \epsilon,$$

which results in the lower bound in (38). In order to upper bound the probability we pick $\boldsymbol{i}^* = i$, we notice that

$$\Pr_{\substack{\boldsymbol{t}_1,\ldots,\boldsymbol{t}_n \\ \sim \text{Exp}(1)}}[\boldsymbol{i}^* = i] \leq \frac{|x_i|}{\|x\|_1} + \Pr_{\substack{\boldsymbol{t}'=(\boldsymbol{t}'_1,\ldots,\boldsymbol{t}'_n) \\ \boldsymbol{t}'_i \sim \text{Exp}(|x_i|)}}\left[D_{\boldsymbol{t}'}(1) \neq i \wedge \frac{1}{\boldsymbol{t}'_{D_{\boldsymbol{t}'}(1)}} - \frac{1}{\boldsymbol{t}'_i} \leq 2\epsilon\|x\|_1\right],$$

and

$$\begin{aligned}
&\Pr_{\substack{\boldsymbol{t}'=(\boldsymbol{t}'_1,\ldots,\boldsymbol{t}'_n) \\ \boldsymbol{t}'_i \sim \text{Exp}(|x_i|)}}\left[D_{\boldsymbol{t}'}(1) \neq i \wedge \frac{1}{\boldsymbol{t}'_{D_{\boldsymbol{t}'}(1)}} - \frac{1}{\boldsymbol{t}'_i} \leq 2\epsilon\|x\|_1\right] \\
&\leq \sum_{j \in [n]\setminus\{i\}} \int_{r:0}^{\frac{1}{2\epsilon\|x\|_1}} |x_j| \exp\left(-r \sum_{j' \in [n]\setminus\{i\}} |x_{j'}|\right) \Pr_{\boldsymbol{t}'_i \sim \text{Exp}(|x_i|)}\left[\frac{r}{1 - 2\epsilon\|x\|_1 r} \geq \boldsymbol{t}'_i \geq r\right] dr + \exp\left(-\frac{1}{2\epsilon}\right),
\end{aligned}$$

where the first term bounds the probability that $\frac{1}{\boldsymbol{t}'_i}$ is not the maximum value, but is large enough to be corrupted in $\tilde{\boldsymbol{z}}$ to appear as the maximum, and the second term bounds the probability that the

maximum value is less than $2\epsilon\|x\|_1$ (in which case a corruption index i may make it the maximum). Then, we have

$$\begin{aligned}
&\sum_{j\in[n]\setminus\{i\}} \int_{r:0}^{\frac{1}{2\epsilon\|x\|_1}} |x_j| \exp\left(-r \sum_{j'\in[n]\setminus\{i\}} |x_{j'}|\right) \mathop{\mathbf{Pr}}_{\boldsymbol{t}_i'\sim\mathrm{Exp}(|x_i|)} \left[\frac{r}{1-2\epsilon\|x\|_1 r} \geq \boldsymbol{t}_i' \geq r\right] dr \\
&\quad\leq \sum_{j\in[n]\setminus\{i\}} \int_{r:0}^{\frac{1}{4\epsilon\|x\|_1}} |x_j| \exp\left(-r\|x\|_1\right) \left(1-\exp\left(-\frac{2\epsilon\|x\|_1|x_i|r^2}{1-2\epsilon\|x\|_1 r}\right)\right) dr + \exp\left(-\frac{1}{4\epsilon}\right) \\
&\quad\lesssim \epsilon\|x\|_1\cdot|x_i| \sum_{j\in[n]\setminus\{i\}} |x_j| \int_{r:0}^{\frac{1}{4\epsilon\|x\|_1}} r^2 \exp\left(-r\|x\|_1\right) dr + \exp\left(-\frac{1}{4\epsilon}\right) \lesssim \frac{\epsilon|x_i|}{\|x\|_1} + \exp\left(-\frac{1}{4\epsilon}\right),
\end{aligned}$$

where in the second inequality, we have $1-\exp\left(-\frac{2\epsilon\|x\|_1|x_i|r^2}{1-2\epsilon\|x\|_1 r}\right) \lesssim \epsilon\|x\|_1|x_i|r^2$ when $r \leq 1/(4\epsilon\|x\|_1)$. ■

We will also need the following Lemma which bounds the tail ℓ_2 norm of the scaled stream.

Lemma 6.6 (Generalization of Proposition 1 of [JW18])**.** *Fix $n\in\mathbb{N}$ and $\alpha\geq 0$, and let $\{\mathcal{D}_i\}_{i\in[n]}$ be a collection of distributions over $\mathbb{R}$ satisfying*

$$\mathop{\mathbf{Pr}}_{\boldsymbol{y}\sim\mathcal{D}_i}[|\boldsymbol{y}|\geq t] \leq \frac{\alpha}{t} \qquad \text{for all } i\in[n] \text{ and all } t\geq 1.$$

For any fixed vector $x\in\mathbb{R}^n$ and integer $\beta\geq 1$, consider the random vector $\boldsymbol{z}\in\mathbb{R}^n$ given by letting

$$\boldsymbol{z}_i \stackrel{\text{def}}{=} \boldsymbol{t}_i\cdot x_i, \qquad \text{where } \boldsymbol{t}_i\sim\mathcal{D}_i \text{ independently for all } i\in[n].$$

Then, $\|\boldsymbol{z}_{-2\beta}\|_2 \leq 12\alpha\|x_{-\beta}\|_1/\sqrt{\beta}$ and $\|\boldsymbol{z}_{-\beta}\|_1 \leq 9\alpha\|x\|_1\lceil\log_2 n\rceil$ with probability at least $1-3e^{-\beta/4}$ over the draws of $\boldsymbol{t}_i\sim\mathcal{D}_i$.[20]

Proof: Define the random sets

$$\mathbf{I}_k = \left\{i\in[n] : \frac{\alpha\|x\|_1}{2^{k+1}} \leq |\boldsymbol{z}_i| \leq \frac{\alpha\|x\|_1}{2^k}\right\} \qquad \text{for } k=0,1,2,\ldots,\lceil\log_2 n\rceil,$$

and notice that, for any $i\in[n]$,

$$\mathop{\mathbf{Pr}}_{\boldsymbol{t}_i\sim\mathcal{D}_i}[i\in\mathbf{I}_k] \leq \mathop{\mathbf{Pr}}_{\boldsymbol{t}_i\sim\mathcal{D}_i}\left[|\boldsymbol{t}_i| \geq \frac{\alpha\|x\|_1}{2^{k+1}|x_i|}\right] \leq \frac{2^{k+1}|x_i|}{\|x\|_1} \qquad \text{implying} \qquad \mathop{\mathbf{E}}_{\boldsymbol{t}_1,\ldots,\boldsymbol{t}_n}[|\mathbf{I}_k|] \stackrel{\text{def}}{=} \mu_k \leq 2^{k+1}.$$

Let $\boldsymbol{\mathcal{E}}_1$ denote the event that there exists $k\geq\lceil\log_2(\beta/4)\rceil$ such that $|\mathbf{I}_k| > 4\cdot 2^{k+1}$. Since all draws $\boldsymbol{t}_i\sim\mathcal{D}_i$ are independent, we let $\delta_k = \frac{4\cdot 2^{k+1}}{\mu} - 1 \geq 3$,

$$\begin{aligned}
\mathop{\mathbf{Pr}}_{\boldsymbol{t}_1,\ldots,\boldsymbol{t}_n}[\boldsymbol{\mathcal{E}}_1] &\leq \sum_{k=\lceil\log_2(\beta/4)\rceil}^{\lceil\log_2 n\rceil} \mathop{\mathbf{Pr}}_{\boldsymbol{t}_1,\ldots,\boldsymbol{t}_n}\left[|\mathbf{I}_k| > 4\cdot 2^{k+1}\right] = \sum_{k=\lceil\log_2(\beta/4)\rceil}^{\lceil\log_2 n\rceil} \mathop{\mathbf{Pr}}_{\boldsymbol{t}_1,\ldots,\boldsymbol{t}_n}[|\mathbf{I}_k| > (1+\delta_k)\mu_k] \\
&\leq \sum_{k=\lceil\log_2(\beta/4)\rceil}^{\lceil\log_2 n\rceil} \exp\left(-2^k\right) \leq 2\exp(-\beta/4),
\end{aligned}$$

[20] For any vector $x\in\mathbb{R}^n$ and any integer $\beta\geq 1$, recall that we define $x_{-\beta}\in\mathbb{R}^n$ be the vector given by x where the β highest magnitude coordinates are set to 0. When $\beta\geq 1$ is not an integer, $x_{-\beta}$ is interpreted as $x_{-\lfloor\beta\rfloor}$.

by a Chernoff bound. Similarly, every $i \in [n]$ satisfies $|\boldsymbol{z}_i| \geq 4\alpha\|x\|_1/\beta$ with probability at most $\beta|x_i|/(4\|x\|_1)$ over the draw of $\boldsymbol{t}_i \sim \mathcal{D}_i$. The event $\boldsymbol{\mathcal{E}}_2$ that more than β indices $i \in [n]$ satisfy $|\boldsymbol{z}_i| \geq 4\alpha\|x\|_1/\beta$ occurs with probability at most $\exp(-\beta/4)$. Thus, whenever $\boldsymbol{\mathcal{E}}_1$ and $\boldsymbol{\mathcal{E}}_2$ do not occur, (which happens with probability at least $1 - 3e^{-\beta/4}$), we have

$$\|\boldsymbol{z}_{-\beta}\|_2^2 \leq \sum_{k=\lceil \log_2(\beta/4)\rceil}^{\lceil \log_2 n\rceil} |\mathbf{I}_k| \cdot \frac{\alpha^2\|x\|_1^2}{2^{2k}} + n \cdot \frac{\alpha^2\|x\|_1^2}{n^2} \leq \frac{128\alpha^2\|x\|_1^2}{\beta},$$

$$\|\boldsymbol{z}_{-\beta}\|_1 \leq \sum_{k=\lceil \log_2(\beta/4)\rceil}^{\lceil \log_2 n\rceil} |\mathbf{I}_k| \cdot \frac{\alpha\|x\|_1}{2^{k}} + n \cdot \frac{\alpha\|x\|_1}{n} \leq \alpha\|x\|_1\,(8\lceil \log_2 n\rceil + 1)$$

since once $\beta \geq n$, the bound becomes 0. Hence, it follows that $\|\boldsymbol{z}_{-\beta}\|_2 \leq 12\alpha\|x\|_1/\sqrt{\beta}$ with probability at least $1 - 3e^{-\beta/4}$. Consider applying the above argument with the vector $x_{-\beta}$ with $\boldsymbol{z}'_i = (x_{-\beta})_i \cdot \boldsymbol{t}_i$ to bound $\|\boldsymbol{z}'_{-\beta}\|_2 \leq 12\alpha\|x_{-\beta}\|_1/\sqrt{\beta}$, and then note that $\|\boldsymbol{z}_{-2\beta}\|_2 \leq \|\boldsymbol{z}'_{-\beta}\|_2$. ■

Remark 10. *Using Lemmas 6.5 and 6.6, along with the standard Count-Sketch of Theorem 11, one obtains a $O(\log^3(n))$ bits of space algorithm for ℓ_1 sampling with relative error $(1/\log n)$ and additive error $1/\mathrm{poly}(n)$ which never outputs **FAIL**. This can be naturally extended to $p \in (0, 2]$, where the space increases by a $\log(n)$ factor for $p = 2$. While this is still much weaker than the perfect sampler of [JW18], it matches the complexity of the best known approximate sampler of [JST11] prior to the results of [JW18], which yields the same space and error guarantee. The advantage of the algorithm resulting from Lemmas 6.5 is that it is perhaps simpler to state and present, and may be useful for pedagogic purposes.*

6.3 Sketching Tools

In this section, we describe some classical tools from the sketching literature to obtain approximations $\tilde{z}$ of a given vector $z \in \mathbb{R}^n$ via a small space linear sketch $\mathbf{S}z$.

Theorem 11 (Count-Sketch [CCFC02])**.** *Fix any $\epsilon > 0$ and $n \in \mathbb{N}$, and let $k = O(\log n/\epsilon^2)$. There is a distribution $\mathcal{S}_k^\infty(n, \epsilon)$ supported on pairs $(\mathbf{S}, \mathrm{Alg}_{\mathbf{S}}^\infty)$, where $\mathbf{S} \in \mathbb{R}^{k\times n}$ is a matrix encoded with $O(k \log n)$ bits of space, and $\mathrm{Alg}_{\mathbf{S}}^\infty$ is an algorithm which receives a vector in $\mathbb{R}^k$ and outputs a vector in $\mathbb{R}^n$. For any $x \in \mathbb{R}^n$,*

$$\|x - \mathrm{Alg}_{\mathbf{S}}^\infty(\mathbf{S}x)\|_\infty \leq \epsilon\|x_{-1/\epsilon^2}\|_2$$

with probability at least $1 - 1/\mathrm{poly}(n)$ over the draw of $(\mathbf{S}, \mathrm{Alg}_{\mathbf{S}}^\infty) \sim \mathcal{S}_k^\infty(n, \epsilon)$.

For notational simplicity, notice that count-sketch may be applied to an $n \times m$ matrix $\mathbf{X}$, by multiplying $\mathbf{SX}$ and applying $\mathrm{Alg}_{\mathbf{S}}$ to each column of $\mathbf{SX}$. If $m \leq \mathrm{poly}(n)$, we may apply a union bound over all m columns and obtain analogous point-wise estimates of all entries of $\mathbf{X}$. The error obtained for an entry $\mathbf{X}_{i,j}$ naturally depends on the (tail)-ℓ_2 norm of the j-th column of $\mathbf{X}$ as per the above theorem. Our algorithm will use the count-sketch of Theorem 11, but also a more general version will be needed.

Theorem 12 (Nested Count-Sketch)**.** *Let $n, m, k \in \mathbb{N}$, $\eta \in (0,1)$, and consider a partition $U = \{U_\ell\}_{\ell \in [m]}$ of $[n]$, $k \leq m$, as well as $s = O(k(\log n)^2/\eta^2)$. There exists a distribution $\mathcal{S}_k^{\infty,U}(n,\eta)$ supported on pairs $(\mathbf{S}, \mathrm{Alg}_{\mathbf{S}}^{\infty,U})$, where $\mathbf{S} \in \mathbb{R}^{s \times n}$ is a matrix encoded with $O(s \log n)$ bits of space, and $\mathrm{Alg}_{\mathbf{S}}^{\infty,U}$ is an algorithm which receives a vector in $\mathbb{R}^s$ and outputs a vector in $\mathbb{R}^n$. Fix any vector $x \in \mathbb{R}^n$, any subset $J \subset [m]$ of size at most k, and any $j \in J$, and let $y \in \mathbb{R}^n$ be the vector given by letting for $i \in [n]$,*

$$y_i = \begin{cases} 0 & i \in \bigcup_{\ell \in J \setminus \{j\}} U_\ell \\ x_i & o.w. \end{cases}.$$

Then, with probability at least $1 - 1/\mathrm{poly}(n)$ over the draw of $(\mathbf{S}, \mathrm{Alg}_{\mathbf{S}}^{\infty,U}) \sim \mathcal{S}_k^{\infty,U}(n,k,\eta)$, every $i \in U_j$ satisfies

$$\left| x_i - \mathrm{Alg}_{\mathbf{S}}^{\infty,U}(\mathbf{S}x)_i \right| \leq \eta \|y_{-1/\eta^2}\|_2.$$

Proof: Let $\boldsymbol{h}\colon [m] \to [10k]$ be a 4-wise independent hash function. For each $t \in [10k]$, we let $\boldsymbol{z}^{(t)} \in \mathbb{R}^n$ be the vector given by setting, for each $i \in [n]$

$$\boldsymbol{z}_i^{(t)} = \begin{cases} x_i & i \in U_j \text{ where } \boldsymbol{h}(j) = t \\ 0 & \text{o.w.} \end{cases}.$$

Instantiate a Count-Sketch data structure by sampling from $(\mathbf{S}_t, \mathrm{Alg}_{\mathbf{S}_t}^{\infty}) \sim \mathcal{S}_k^{\infty}(n,\eta)$, and storing $\mathbf{S}_t \boldsymbol{z}^{(t)}$. Letting $\boldsymbol{t} = \boldsymbol{h}(j)$, notice that whenever $\boldsymbol{h}(j') \neq \boldsymbol{t}$ for all $j' \in J \setminus \{j\}$ (which occurs with probability at least $(1 - 1/(10k))^{k-1} \geq 0.9$), then $\boldsymbol{z^{(t)}} = y$. Hence, the algorithm $\mathrm{Alg}_{\mathbf{S}}^{\infty}(\mathbf{S_t} \boldsymbol{z^{(t)}})$ recovers y up to ℓ_∞ error at most $\eta \|y_{-1/\eta^2}\|_2$. We note that the nested application of $\boldsymbol{h}$ and $\mathbf{S}_1, \ldots, \mathbf{S}_{10k}$ may be represented as a matrix multiplication of x, and we may boost error probability by repeating $O(\log n)$ times and taking the median.

■

We will sometimes use the following shorthand notation to represent the error yielded from Theorem 12.

Definition 6.7. *Given a dimension n, $t \geq 0$, and a partition $U = \{U_\ell\}_{\ell \in [m]}$ of $[n]$, let $x \in \mathbb{R}^n$. Let $J \subset [m]$ be a subset of the partition indices. Then we write $x_{-(J,t)} = (y^J)_{-t}$, where y^J is defined via*

$$y_i^J = \begin{cases} x_i & \text{if } i \notin \cup_{\ell \in J} U_\ell \\ 0 & \text{otherwise} \end{cases}$$

We will need to open the black-box for the ℓ_1-sketch cited in Theorem 7.

Theorem 13 (ℓ_1-sketch [Ind06a])**.** *Fix $n \in \mathbb{N}$, $\epsilon, \delta \in (0,1)$, and $k = O(\log(1/\delta)/\epsilon^2)$. Let $\mathcal{S}_k^1(n)$ be the distribution on $k \times n$ matrices $\boldsymbol{\Omega}$ with independent Cauchy random variables $\mathcal{C}$, and let Alg^1 be the algorithm which takes a vector $y \in \mathbb{R}^k$, and outputs*

$$\mathrm{Alg}^1(y) \stackrel{\text{def}}{=} \text{median}\left\{ \frac{|y_i|}{\text{median}(|\mathcal{C}|)} : i \in [k] \right\}, \quad \textit{where} \quad \text{median}(|\mathcal{C}|) \stackrel{\text{def}}{=} \sup\left\{ t : \Pr_{\boldsymbol{\omega} \sim \mathcal{C}}[|\boldsymbol{\omega}| \leq t] \leq \frac{1}{2} \right\}.$$

Then, for any vector $x \in \mathbb{R}^n$,

$$\Pr_{\boldsymbol{\Omega} \sim \mathcal{S}_k^1(n)} \left[\left| \mathrm{Alg}^1(\boldsymbol{\Omega}x) - \|x\|_1 \right| \leq \epsilon \|x\|_1 \right] \geq 1 - \delta.$$

6.4 Construction of the Sketch

Table 1: Table of Notation

η	$\triangleq$	precision for Count-Sketch
ρ	$\triangleq$	$\Theta(\frac{1}{\epsilon_0^2}\log s)$ number of Cauchy Sketches
$\mathbf{X}$	$\triangleq$	matrix Encoding parents in $\mathcal{V}_{i-1}$
$\mathbf{Y}$	$\triangleq$	matrix Encoding children in $\mathcal{V}_i$
Δ_i	$\triangleq$	$\sum_{v\in\mathcal{V}_i} \lvert\lvert A_v\rvert - \lvert B_v\rvert\rvert$
$\Delta_i(u)$	$\triangleq$	$\sum_{v\,:\,\pi(v)=u} \lvert\lvert A_v\rvert - \lvert B_v\rvert\rvert$
λ_u	$\triangleq$	vector with coordinates $\lambda_{u,v} = \lvert A_v\rvert - \lvert B_v\rvert$
$\omega_{u,j}$	$\triangleq$	vector of i.i.d. Cauchy random variables
$\mathbf{\Omega}_u$	$\triangleq$	$\rho \times d$ matrix of i.i.d. Cauchy random variables
$\mathbf{D}^1, \mathbf{D}^2$	$\triangleq$	Diagonal exponential scaling matrices
$\mathbf{Z}^1$	$\triangleq$	$\mathbf{D}^1\mathbf{X}$
$\mathbf{Z}^2$	$\triangleq$	$\mathbf{D}^2\mathbf{Y}$
$\mathbf{S}^1$	$\triangleq$	Count-Sketch Matrix from Theorem 11
$\mathbf{S}^2$	$\triangleq$	*Nested* Count-Sketch Matrix from Theorem 12
$\tilde{\mathbf{Z}}^j$	$\triangleq$	estimate of $\mathbf{Z}^j$ from Count-Sketch and Nested Count-Sketch

Remark 14. *In order to ease the presentation, in this section and the one follows (Section 6.5), we will use boldface letters to denote (possibly random) matrices, and drop the boldface letters for random variables.*

Fix any $i \in [h]$, we now describe the sketching algorithm to estimate $\mathcal{I}_i$. We include a table of notation used in this section and the following in Figure 1. As we will see, we will not always be able to obtain a good approximation to $\mathcal{I}_i$ for all levels i. Specifically, for the levels i for which $\mathcal{I}_i$ is not sufficiently large, we will not obtain an estimate of $\mathcal{I}_i$, however we will be able to safely ignore such "small" levels. Moreover, we will be able to easily determine when a level is small, and avoid obtaining an estimate for it. Thus, we now describe the construction for the sketch for level i. Recall that the input to the $\mathrm{EMD}(A, B)$ sketching problem can be represented as a vector $f_{A,B} \in \mathbb{R}^{2\cdot 2^d}$, where the first 2^d coordinates describe the multi-set $A \subset \{0,1\}^d$ and the last 2^d coordinates describe $B \subset \{0,1\}^d$. We design a linear sketch which can be represented as $\mathbf{S} \cdot f_{A,B}$ for some matrix $\mathbf{S}$.

As discussed earlier, our approach will be to first sample a vertex v^* from $\mathcal{V}_i$, such that

$$\mathbf{Pr}\left[v^* = v\right] \tilde{\propto} \lvert\lvert A_v\rvert - \lvert B_v\rvert\rvert,$$

where $\tilde{\propto}$ is meant to indicate that the relationship is approximately proportional (since we encounter some errors). Letting u^* be the parent of v^*, the sketch will estimate q_{u^*,v^*}. If we can produce

polylog(s, d) samples from such a distribution and obtain estimates of q_{u^*,v^*}, the empirical mean will produce an estimate to $\mathcal{I}_i$. We design a linear data-structured based on precision sampling that will allow us to both sample v^* and simultaneously recover an estimate of q_{u^*,v^*}. To do this, we define two matrices $\mathbf{X}, \mathbf{Y}$ whose entries will be linear combinations of the entries of f. The first matrix $\mathbf{X}$ encodes information about the nodes in the level $\mathcal{V}_{i-1}$, and $\mathbf{Y}$ will encode information about the children nodes in $\mathcal{V}_i$.

The rows of $\mathbf{X}$ will be indeed by nodes $u \in \mathcal{V}_{i-1}$, and for each row $\mathbf{X}_u$ we will store several pieces of information: **(1)** an approximation of the total discrepancy of its children, namely $\sum_{v:\pi(v)=u} ||A_v| - |B_v||$, and **(2)** sufficient meta-data to compute the center c_u. Similarly, in $\mathbf{Y}$, each row is indexed by a vertex $v \in T_i$, and will store the discrepancy $||A_v| - |B_v||$ at that node, as well as sufficient information to compute the center c_v. The complete details now follow.

Let η be the precision parameter for count-sketch which we will soon fix, and let ϵ_0 be another precision parameter for Cauchy-sketches that set to $\epsilon_0 = \Theta(1/\log s)$. In what follows, let Δ_i be the total discrepancy at level i, namely $\Delta_i = \sum_{v \in \mathcal{V}_i} ||A_v| - |B_v||$. For each parent node $u \in \mathcal{V}_{i-1}$, define the vector $\lambda_u \in \mathbb{R}^{2^{2^i}}$ indexed by the children of u. Specifically, for each $v \in \mathcal{V}_i$ with $\pi(v) = u$, we have an entry $\lambda_{u,v}$, with the value $\lambda_{u,v} = |A_v| - |B_v|$. Thus, λ_u is the vector of discrepancies at u, and we have $\sum_{u \in \mathcal{V}_{i-1}} \|\lambda_u\|_1 = \Delta_i$.

Construction of the Matrices. We first define the matrix $\mathbf{X} \in \mathbb{R}^{|\mathcal{V}_i| \times D_1}$, where $D = 2\rho + 1$, and $\rho = \Theta(\epsilon_0^{-2} \log s)$. Each row, denoted $\mathbf{X}_u$, of $\mathbf{X}$ corresponds to a unique vertex $u \in \mathcal{V}_{i-1}$ (i.e., the rows of $\mathbf{X}$ are indexed by vertices in $\mathcal{V}_{i-1}$). Let $\mathbf{X}_{u,i}$ denote the i-th entry in the v-th row of A. Ideally, we would like to set $\mathbf{X}_{u,1} = \sum_{v:\pi(v)=u} ||A_v| - |B_v|| = \|\lambda_u\|_1$. However, this will not be possible to do in the linear sketching setting, because $\|\lambda_u\|_1$ cannot be expressed as a linear function of the input vector f. Instead, we will try to approximate this value ℓ_1 norm of $\|\lambda_u\|_1$ with a Cauchy sketch. Specifically, for $i = 1, 2, \ldots, \rho$, we generate a vector $\omega_{u,i} \in \mathbb{R}^{2^{2^i}}$ of i.i.d. Cauchy random variables $\mathcal{C}$. Note that the algorithm will only need to actually generate at most $2s$ random variables in the vector $\omega_{u,i}$, and moreover $\omega_{u,i}$ can be stored in small space using limited independence [KNW10]. Next, for $i = 1, 2, \ldots, \rho$, we set $\mathbf{X}_{u,i} = \langle \lambda_u, \omega_{u,i} \rangle$. Note that by Theorem 13, we have $\frac{1}{\text{median}(|\mathcal{C}|)} \text{median}_{i \in [\rho]} |\mathbf{X}_{u,i}| = (1 \pm \epsilon_0)\|\lambda_u\|_1$.

Next, for the remaining $\rho + 1$ coordinates in each row $\mathbf{X}_u$, we set $\mathbf{X}_{u,\rho+1} = |A_u| + |B_u|$, which can be accomplished exactly via an linear combination of the entries in f. Lastly, we generate an i.i.d. Cauchy matrix $\Omega_u \in \mathbb{R}^{\rho \times d}$, and e set the final ρ entries to be $\Omega_u \cdot \left(\sum_{p \in A_u \cup B_u} p \right)$, where $p \in \{0,1\}^d$ is thought of as a column vector. Thus, the information in the row $\mathbf{X}_u$ contains a $(1 \pm \epsilon_0)$ approximation of the total discrepancy at the vertex u, the number of points in $|A_u \cup B_u|$, and the sum of the points in u after being multiplied by a Cauchy sketch.

We now describe the construction of the matrix $\mathbf{Y}$, which will be similar but simpler since we will not need the Cauchy sketch for the discrepancies. For each $v \in \mathcal{V}_i$, we similarly have a assigned row $\mathbf{Y}_v$. We set $\mathbf{Y}_{v,1} = ||A_v| - |B_v||$, and we set $\mathbf{Y}_{v,2} = |A_v| + |B_v|$, and the last ρ coordinates of $\mathbf{Y}_v$ will be set to $\Omega_{\pi(v)} \cdot \left(\sum_{p \in A_v \cup B_v} p \right)$. In summary, we have

$$\mathbf{X}_u = \left[\langle \lambda_u, \omega_{u,1}\rangle, \quad \dots, \quad \langle \lambda_u, \omega_{u,\rho}\rangle, \quad |A_u| + |B_u|, \quad \left(\Omega_u \cdot \sum_{p \in A_u \cup B_u} p\right)_1, \quad \dots, \quad \left(\Omega_u \cdot \sum_{p \in A_u \cup B_u} p\right)_\rho\right]$$

$$\mathbf{Y}_v = \left[||A_v| - |B_v||, \quad |A_v| + |B_v|, \quad \left(\Omega_{\pi(v)} \cdot \sum_{p \in A_v \cup B_v} p\right)_1, \quad \dots, \quad \left(\Omega_{\pi(v)} \cdot \sum_{p \in A_v \cup B_v} p\right)_\rho\right]$$

Construction of the sketch. The sketch is now as follows. We generate two count-sketch matrices $\mathbf{S}^1 \in \mathbb{R}^{k \times n_1}, \mathbf{S}^2 \in \mathbb{R}^{k' \times n_2}$ with precision parameter η, where $\mathbf{S}^1 \sim \mathcal{S}_k^\infty(n_1, \eta)$ is a standard count-sketch matrix from Theorem 11, and $\mathbf{S}^2 \sim \mathcal{S}_k^{\infty,U}(n_2, \eta)$ is a *nested-count sketch* from Theorem 12 (applied with $\epsilon = \epsilon_0 = \Theta(1/\log s)$), so that $k = O(1/\eta^2 \log(n_1 + n_2))$ and $k' = O(\eta^{-2}\log(n_1 + n_2)\log s)$, and where $n_1 = |\mathcal{U}_1| = \tilde{O}(s^2)$ and $n_1 = |\mathcal{U}_1 \times \mathcal{U}_2| = \tilde{O}(s^4)$. We also generate two diagonal precision scaling matrices $\mathbf{D}^1, \mathbf{D}^2$, where $\mathbf{D}^1_{u,u} = 1/t^1_u$, and $D^2_{v,v} = \frac{1}{t^1_{\pi(v)} t^2_v}$, where $t^j_i \sim \text{Exp}(1)$ are independent exponential random variables. In the sequel, we will drop the superscript and write t_v for some vertex v, since v will be in exactly one of $\mathcal{V}_i$ or $\mathcal{V}_{i-1}$ it will be clear whether t_v was from the first or second set of exponential variables. Next, we store a Cauchy sketch $\mathbf{\Omega Y}_{*,1}$ with $O(\log(s)/\epsilon_0^2)$ rows, which by Theorem 13 gives a estimate $\tilde{s} = (1 \pm \epsilon_0)\Delta_i$ with probability $1 - 1/\text{poly}(s)$. Altogether, our algorithm stores the three linear sketches $\mathbf{S}^1\mathbf{D}^1\mathbf{X}$, $\mathbf{S}^2\mathbf{D}^2\mathbf{Y}$, and $\mathbf{\Omega Y}_{*,1}$. To define how the error for the nested-count sketch $\mathbf{S}^2\mathbf{D}^2\mathbf{Y}$ is applied, we need to define a partition of the rows of $\mathbf{Y}$, which we do naturally by partitioning $\mathcal{V}_i$ into the subsets of children v which share the same parent in $u \in \mathcal{V}_{i-1}$. Notice that if $\tilde{\mathbf{Z}}^2$ is the estimate of $\mathbf{D}^2\mathbf{Y}$ produced by the nested count-sketch, we have for any fixed subset $U \subset \mathcal{V}_{i-1}$ of size $|U| = 1/\epsilon_0$, for any $v \in \mathcal{V}_i$ we have

$$\|\tilde{\mathbf{Z}}^2_{v,j} - (\mathbf{D}^2\mathbf{Y})_{v,j}| \leq \eta \|((\mathbf{D}^2\mathbf{Y})_{*,j})_{-(U \setminus \pi(v), 1/\eta^2)}\|_2$$

where for a vector x, $(x)_{-(U,c)}$ is defined as in Theorem 12. Notice that $\|((\mathbf{D}^2\mathbf{Y})_{*,j})_{-(U,1/\eta^2)}\|_2 \leq \|((\mathbf{D}^2\mathbf{Y})_{*,j})_{-1/\eta^2}\|_2$, so we will sometimes use this weaker bound when the tighter one is not required.

6.5 Analysis of the Algorithm.

We begin this section with a table of notation for reference, followed by the full algorithm. The algorithm proceeds in two main steps. First, it samples a parent $u^* \sim \mathcal{V}_{i-1}$ with probability proportional to $\Delta_i(u^*)$. It does this by first scaling a vector whose coordinates are approximations of $\Delta_i(u)$, for $u \in \mathcal{V}_{i-1}$, by independent inverse exponentials. By returning an approximation of the maximum coordinate after scaling, we can apply Lemma 6.5 to demonstrate that the parent u^* obtained is indeed from the correct distribution. We then iterate the process, searching over all children $v \in \mathcal{V}_i$ of u^*, and finally sampling v^* with $\pi(v^*) = u^*$ with probability proportional to $||A_{v^*}| - |B_{v^*}||$, again by the same procedure using Lemma 6.5. Once we have obtained our desired samples (u^*, v^*), we would like to output the quantity $\Delta_i \cdot q_{u^*,v^*}$, where q_{u^*,v^*} is as defined as in Equation 36. We can easily obtain an approximation $\tilde{\Delta}_i$ of Δ_i using a Cauchy Sketch of Lemma 13. To obtain an approximation of q_{u^*,v^*}, we utilize the metadata contained within the rows $\mathbf{X}_{u^*,*}, \mathbf{Y}_{v^*,*}$, of which we have approximations of due to count-sketch.

We prove two main Lemmas which will yield the correctness of our algorithm. First Lemma 6.8 demonstrates that the pair (u^*, v^*) is sampled from the correct distribution. Second, the more involved Lemma 6.9 demonstrates that our approximation of q_{u^*,v^*} is sufficiently accurate. The first demonstrates that the sample $v^* \sim \mathcal{V}_i$ is drawn from the correct distribution. The second demonstrates that conditioned on sampling a $(u^*, v^*) \sim \mathcal{V}_{i-1} \times \mathcal{V}_i$, we can recover good approximations to the centers c_{u^*}, c_{v^*} *if* the parent u^* is not among a small set of "bad parents".

`Procedure to Estimate` $\mathcal{I}_i$
`Inputs` $(\mathbf{S}^1\mathbf{D}^1\mathbf{X}, \mathbf{S}^2\mathbf{D}^2\mathbf{Y})$, an estimate $\tilde{\Delta}_i = (1 \pm \epsilon_0)\Delta_i$.
`Parameters:` $\epsilon > 0$, count-sketch precision $\eta = \Omega\left(\sqrt{\frac{\epsilon}{\log d \log^\ell s}}\right)$ for $\ell = O(1)$, and $\epsilon_0 = \Theta(1/\log s)$.

1. Via $\mathbf{S}^1\mathbf{D}^1\mathbf{X}$ and $\mathbf{S}^2\mathbf{D}^2\mathbf{Y}$ recover approximate matrices $\tilde{\mathbf{Z}}^1, \tilde{\mathbf{Z}}^2$ such that
$$\|\tilde{\mathbf{Z}}^1_{*,j} - (\mathbf{D}^1\mathbf{X})_{*,j}\|_\infty \le \eta\|(\mathbf{D}^1\mathbf{X}_{*,j})_{-1/\eta^2}\|_2, \quad \|\tilde{\mathbf{Z}}^2_{*,j} - (\mathbf{D}^2\mathbf{Y})_{*,j}\|_\infty \le \eta\|(\mathbf{D}^2\mathbf{Y}_{*,j})_{-1/\eta^2}\|_2$$
for all columns j.
2. For each row u of $\tilde{\mathbf{Z}}^1$, compute $\tilde{\Delta}_i(u) = \text{median}_{j\in[\rho]}|\tilde{\mathbf{Z}}^1_{u,j}|$, and $u^* = \arg\max_u \tilde{\Delta}_i(u)$.
3. Compute $v^* = \arg\max_{v:\pi(v)=u^*} |\tilde{\mathbf{Z}}^2_{v,1}|$, and output
$$Q = \frac{\tilde{\Delta}_i}{\text{median}(|\mathcal{C}|)} \cdot \min\left\{\max\left\{\text{median}_j \left|\frac{\tilde{\mathbf{Z}}^1_{u^*,\rho+j+1}}{\tilde{\mathbf{Z}}^1_{u^*,\rho+1}} - \frac{\tilde{\mathbf{Z}}^2_{v^*,j+2}}{\tilde{\mathbf{Z}}^2_{v^*,2}}\right|, \frac{d}{2^i}\right\}, \frac{30d\log s}{2^i}\right\}$$

Figure 5: Main Sketching Primitive to Estimate $\mathcal{I}_i$

We state Lemmas 6.8 and 6.9 below, and proceed to prove our main Theorem 9 given these Lemmas. The proofs of Lemmas 6.8 and 6.9 will then be given in Section 6.5.1.

Lemma 6.8. *Let $v^* \in \mathcal{V}_i$ be the vertex which is sampled in Figure 6.5. Then for any $v \in \mathcal{V}_i$, we have $\mathbf{Pr}[\mathbf{v}^* = v] = (1 \pm \epsilon_0)\frac{||A_v|-|B_v||}{\Delta_i} \pm s^{-c}$, where $c > 1$ is an arbitrarily large constant.*

Lemma 6.9. *Fix η, ϵ_0 as in Figure 6.5, and let (u^*, v^*) be the samples accepted by Figure 6.5. Then assuming that $\frac{2s}{\Delta_i} < \frac{\log s \log d}{\epsilon 2^i}$, and moreover that $\frac{|A_{u^*}|+|B_{u^*}|}{\Delta_i(u^*)} \le \frac{\log s \log d}{\epsilon\nu 2^i}$, where $\nu = \Theta(1/\log s)$, then with probability $1 - 1/\text{poly}(s)$, we have*
$$\frac{1}{\text{median}(|\mathcal{C}|)} \cdot \text{median}_{j\in[\rho]} \left|\frac{\tilde{\mathbf{Z}}^1_{u^*,\rho+j+1}}{\tilde{\mathbf{Z}}^1_{u^*,\rho+1}} - \frac{\tilde{\mathbf{Z}}^2_{v^*,j+2}}{\tilde{\mathbf{Z}}^2_{v^*,2}}\right| = (1 \pm \epsilon_0)\|c_{u^*} - c_{v^*}\|_1 \pm \epsilon_1\frac{d}{2^i}$$
where ϵ_1 is an arbitrarily small constant.

Finally, we will need a brief fact that allows us to disregard a small fraction $u \in \mathcal{V}_{i-1}$ which are in a subset of "bad parents".

Fact 6.10. *Fix $\gamma > 1$ and $\nu \in (0, 1)$, and suppose that we have $\frac{2s}{\Delta_i} < \gamma$. Let $W \subset \mathcal{V}_{i-1}$ be the set of $u \in \mathcal{V}_{i-1}$ such that $\frac{|A_{u^*}|+|B_{u^*}|}{\Delta_i(u)} \le \frac{\gamma}{\nu}$. Then $\sum_{u\in W} \Delta_i(u) \ge (1 - \nu)\Delta_i$*

Proof: Suppose otherwise. Then for each $u \notin W$, we have $\frac{\Delta_i(u)}{||A_{u^*}|+|B_{u^*}||} < \frac{\nu}{\gamma}$, so

$$\frac{\Delta_i}{2s} < \frac{1}{2s\nu}\sum_{u\notin W}\Delta_i(u) \leq \frac{1}{2\gamma s}\sum_{u\notin W}||A_{u^*}|+|B_{u^*}|| \leq \frac{1}{\gamma} \tag{39}$$

Thus $\Delta_i/(2s) < \gamma$, a contradiction. ■

We now restate Theorem 9, and prove it given Lemmas 6.8 and 6.9.

Theorem 9 *For $d, s \in \mathbb{N}$, there exists a 1-round linear sketching algorithm such that given multi-sets $A, B \subset \{0,1\}^d$ with $|A| = |B| = s$, computes an approximate $\widehat{\mathcal{I}}$ to $\mathrm{EMD}(A,B)$ with*

$$\mathrm{EMD}(A,B) \leq \widehat{\mathcal{I}} \leq \tilde{O}(\log s)\,\mathrm{EMD}(A,B) + \epsilon ds$$

with probability at least $3/4$. Moreover, the space used by the linear sketch is $\tilde{O}(1/\epsilon)$.

Proof: Fix any level $i \in [h]$ (where $h = O(\log d)$). First note that setting the Cauchy Sketch $\mathbf{\Omega Y}_{*,1}$ to be derandomized by [KNW10], the matrix $\mathbf{\Omega Y}_{*,1}$ and $\mathbf{\Omega}$ can be stored using $O(1/\epsilon_0^2 \log^2(s))$ bits of space, and yields the approximation $\tilde{\Delta}_i = (1\pm\epsilon_0)\Delta_i = (1\pm\epsilon_0)\|\mathbf{Y}_{*,1}\|_1$ with probability $1 - 1/\mathrm{poly}(s)$. We condition on this correctness now, which is independent of the randomness of the rest of the algorithm, and use the same estimate $\tilde{\Delta}_i$ for all repetitions of the sampling algorithm in Figure 6.5. Now note that if $\frac{2s}{\tilde{\Delta}_i} > \frac{\log s \log d}{4\epsilon 2^i}$, then

$$\mathcal{I}_i \leq \Delta_i \frac{30d\log s}{2^i} \leq O\left(\epsilon ds \frac{1}{\log d}\right)$$

so the total contribution of all such $i \in [\log d]$ is at most $O(\epsilon ds)$, which we absorb into the additive error (after rescaling ϵ by a constant). We can then use our estimate $\tilde{\Delta}_i = (1\pm\epsilon_0)\Delta_i$ to test if this is the case. Specifically, if we have $\frac{2s}{\tilde{\Delta}_i} > \frac{\log s \log d}{2\epsilon 2^i}$, then we can be sure that $\frac{\log s \log d}{\epsilon 2^i} > \frac{2s}{\Delta_i} > \frac{\log s \log d}{4\epsilon 2^i}$, and in this case we set $\widehat{\mathcal{I}}_i = 0$. Otherwise, we will attempt to estimate $\mathcal{I}_i$ via the algorithm in Figure 6.5. Notice that if we happened to choose $\epsilon > \Omega(\frac{\log s \log d}{d})$, then necessarily there will be some levels i (lower in the tree) such that $\frac{2s}{\Delta_i} > \frac{\log s \log d}{4\epsilon 2^i}$ no matter what, since for such a setting of ϵ the right hand side is less than 1. For smaller ϵ, however, every level $i \in [h]$ may have a chance to contribute.

We apply Lemma 6.9 with $\nu = \Theta(1/\log s)$ as small enough constant, and note by Fact 6.10, that the set W of $u \in \mathcal{V}_{i-1}$ that satisfy the assumption of Lemma 6.14 satisfies $\sum_{u\in W}\Delta_i(u) > (1-\nu)\Delta_i$. By the setting of ν and the fact that $q_{u,v}/q_{u',v'} < 30\log s$ for any non-empty $v, v' \in \mathcal{V}_i$, it follows that $\sum_{u\in W}\sum_{v:\pi(v)=u} q_{u,v} = (1 \pm 1/100)\mathcal{I}_i$. By Lemma 6.9, conditioned on sampling $u^* \in W$, our estimate satisfies

$$\frac{1}{\mathrm{median}(|\mathcal{C}|)} \cdot \mathrm{median}_{j\in[\rho]}\left|\frac{\tilde{\mathbf{Z}}^1_{u^*,\rho+j+1}}{\tilde{\mathbf{Z}}^1_{u^*,\rho+1}} - \frac{\tilde{\mathbf{Z}}^2_{v^*,j+2}}{\tilde{\mathbf{Z}}^2_{v^*,2}}\right| = (1\pm\epsilon_0)\|c_{u^*} - c_{v^*}\|_1 \pm \epsilon_1\frac{d}{2^i}$$

with probability $1 - s^{-c}$. Conditioned on this, and using that $q_{\pi(v),v} \geq d/2^i$ for all $v \in \mathcal{V}_i$ by definition, we then have $Z = (1\pm 8\epsilon_1)\Delta_i \cdot q_{u^*,v^*}$. Thus the expectation of Z conditioned on sampling (u^*, v^*) with $u^* \in W$ is $(1\pm 8\epsilon_1)\Delta_i(q_{u^*,v^*} \pm s^{-c}(30d\log s/2^i)) = (1 \pm 10\epsilon_1)\Delta_i q_{u^*,v^*}$. Moreover, by Lemma 6.8, we know that $v^* = v$ with probability $(1\pm\epsilon_0)||A_v| - |B_v||/\Delta_i \pm s^{-c}$. It follows that

the probability that we sample $v^* = v \in \mathcal{V}_i$ with $\pi(v) \notin W$ is at most $(3/2)\nu$. Putting these facts together, we have that

$$\begin{aligned}
\mathbf{E}[Q] &= \left(\sum_{u \in W} \sum_{v:\pi(v)=u} (1 \pm \epsilon_0) \left(\frac{||A_v| - |B_v||}{\Delta_i} \pm s^{-c} \right) (1 \pm 10\epsilon_1)\Delta_i \cdot q_{u,v} \right) \pm 60\nu\Delta_i \frac{d \log s}{2^i} \\
\mathbf{E}[Q] &= (1 \pm 12\epsilon_1) \left(\sum_{u \in W} \sum_{v:\pi(v)=u} ||A_v| - |B_v|| \boldsymbol{q}_{u,v} \right) \pm \left(60\nu\Delta_i \frac{d \log s}{2^i} + s^{-c+3} \frac{d}{2^i} \right) \qquad (40) \\
&\leq (1 \pm 1/50))\mathcal{I}_i \pm \Delta_i \frac{d}{100 \cdot 2^i} \\
&\leq (1 \pm \frac{1}{20})\mathcal{I}_i
\end{aligned}$$

where we used that $\mathcal{I}_i \geq \Delta_i \frac{d}{2^i}$ by definition. Similarly, since $Q \leq \Delta_i \frac{20d \log s}{2^i}$ (conditioned on the high probability success of our estimate for Δ_i), it follows that $\mathbf{Var}[Q] < (\Delta_i \frac{20d \log s}{2^i})^2$. Thus, repeating the procedure $O(\log^2 s)$ times and obtaining $Q_1, Q_2, \ldots, Q_{O(\log^2 s)}$, we have that $\sqrt{\mathbf{Var}\left[\sum_{j=1}^{O(\log^2 s)} Q_j\right]} < \Delta_i \frac{d}{100 \cdot 2^i} \leq \mathcal{I}_i/100$. By Chebyshev's inequality, we have $\sum_{j=1}^{O(\log^2 s)} Q_j = (1 \pm 1/10)\mathcal{I}_i$ with probability at least 99/100. Thus, after scaling $\widehat{\mathcal{I}}_i$ up by a factor of $(1 + 1/3)$, we can set $\widehat{\mathcal{I}} = \sum_i \widehat{\mathcal{I}}_i + O(\epsilon s)$, and after scaling ϵ down by a constant we have $\mathcal{I} \leq \mathcal{I}_i \leq 2\mathcal{I} + \epsilon s$. By Remark 5, we know that $\mathrm{EMD}(A, B) \leq \mathcal{I} \leq \tilde{O}(\log s)\,\mathrm{EMD}(A, B)$ with probability at least .89 over the draw of the tree, which completes the proof of the claimed approximation after a union bound over all aforementioned events. To see the space complexity, note for each of the $h = O(\log d)$ levels, and for each of the $\tilde{O}(1)$ samples used in that level, we the size of the count-sketch and nested count-sketch is $\tilde{O}(\eta^{-2})$ by Theorems 11 and 12. Notice that to apply Theorems 11 and 12, we used the fact that, after the universe reduction step from Proposition 5.1, both $\mathbf{X}, \mathbf{Y}$ have at most $\mathrm{poly}(s)$ rows. This completes the proof after noting that $\eta = \tilde{O}(\sqrt{\epsilon/d})$. ■

A Streaming Algorithm. Next, we demonstrate how this linear sketching algorithm results in a one-pass streaming algorithm for the turnstile model. Recall in this model, a sequence of at most $\mathrm{poly}(s)$ updates arrives in the stream, where each update either inserts or deletes a point $p \in \{0, 1\}^d$ from A, or inserts or deletes a point from B. Noticed that the t-th update can be modeled by coordinate-wise updates to $f_{A,B}$ of the form $(i_t, \Delta_t) \in [2 \cdot 2^d] \times \{-1, 1\}$, causing the change $(f_{A,B})_{i_t} \leftarrow (f_{A,B})_{i_t} + \Delta_t$. At the end of the stream, we are promised that $f_{A,B}$ is a valid encoding of two multi-sets $A, B \subset \{0, 1\}^d$ with $|A| = |B| = s$.

Corollary 6.11. *For $d, s \in \mathbb{N}$, there exists a one-pass turnstile streaming algorithm which, on a stream vector $f_{A,B}$ encoding multi-sets $A, B \subset \{0, 1\}^d$ with $|A| = |B| = s$, the algorithm then computes an approximate $\widehat{\mathcal{I}}$ to $\mathrm{EMD}(A, B)$ with*

$$\mathrm{EMD}(A, B) \leq \widehat{\mathcal{I}} \leq \tilde{O}(\log s)\,\mathrm{EMD}(A, B)$$

with probability at least $3/4$, and uses $O(d + 1/\epsilon) \cdot \mathrm{polylog}(s, d)$ bits of space.

Proof: As in Corollary F.1, we explicitly store the $O(d \log d)$ bits of randomness required to specify the entire Quadtree T, which we now fix. Also, as in the proof of Corollary F.1, we use $O(d)$

bits of space to generate the 2-wise independent hash functions h_i, h_{i-1} needed for the universe reduction step of Proposition 5.1. Moreover, the randomness used in the Cauchy sketch $\mathbf{\Omega Y}_{*,1}$ can be derandomized by the results of [KNW10] to use only $O(1/\epsilon_0^2 \log^2 s)$ bits of space and succeed with probability $1 - 1/\text{poly}(s)$.

We now note that there are several other sources of randomness that must be deranndomzied: namely the parent and child exponentials $\{t_u\}_{u \in \mathcal{V}_{i-1}}, \{t_v\}_{v \in \mathcal{V}_i}$, the Cauchy sketches $\{\Omega_u\}_{u \in \mathcal{V}_{i-1}}$, and $\{\omega_{u,j}\}_{u \in \mathcal{V}_{i-1}, j \in [\rho]}$. We remark that the randomness needed for the count-sketch and nested count sketch matrices $\mathbf{S}^1, \mathbf{S}^2$ of Theorems 11 and 12 use 4-wise independent hash functions, and therefore do not need to be derandomized. We first remark that we can apply a standard truncation of these continuous distributions to be stored in $O(\log sd)$ bits of space (by simply only storing generating them to $O(\log sd)$ bits of accuracy). This introduced a $1/\text{poly}(sd)$ additive error into each coordinate of the linear sketch. First note that this additional error can be absorbed into the error of the count sketch estimates $\tilde{\mathbf{Z}}^1, \tilde{\mathbf{Z}}^2$, increasing the error bounds by at most a $(1 + 1/\text{poly}(sd)))$ factor. The analysis from Lemmas 6.8 and 6.9 apply with this (slightly) larger error after scaling the precision parameter η by the same factor.

We demonstrate how to derandomize these (now discrete) distributions. We adapt a standard argument due to Indyk [Ind06b] based on Nisan's PRG [Nis92]. Since the algorithm is a linear sketch, we can reorder the stream so that for every $u \in \mathcal{V}_{i-1}$ all updates to points $x \in \{0,1\}^d$ with $v_{i-1}(x) = u$ occur in a consecutive block B_u of updates, and B_u is broken up further into blocks B_v for all children v of u, such that all updates to points $x \in \{0,1\}^d$ with $v_i(x) = v$ occur in the consecutive block B_v. We now describe a small space tester which computes the output of our algorithm while reading the randomness for the exponentials $\{t_u\}_{u \in \mathcal{V}_{i-1}}, \{t_v\}_{v \in \mathcal{V}_i}$, the Cauchy sketches $\{\mathbf{\Omega}_u\}_{u \in \mathcal{V}_{i-1}}$, and $\{\omega_{u,j}\}_{u \in \mathcal{V}_{i-1}, j \in [\rho]}$, in a stream. When the block B_u begins for any $u \in \mathcal{V}_{i-1}$, the algorithm reads and stores the randomness needed for $t_u, \mathbf{\Omega}_u$ and $\{\omega_{u,j}\}_{j \in [\rho]}$, which is a total of $\tilde{O}(d\rho) = \tilde{O}(d)$ bits of space. Within the block B_u, when the updates for a block B_v begin for any child v of u, we also store the random variable t_v. We then can fully process all updates to the linear sketches $\mathbf{S}^1\mathbf{D}^1\mathbf{X}, \mathbf{S}^2\mathbf{D}^2\mathbf{Y}$ due to the updates in B_v in space $\tilde{O}(d)$ using only the randomness stored within the blocks B_u, B_v.

After the block B_v is complete, we can discard t_v, since it will never be needed for to process any other update. Similarly, once B_u is complete, we can discard the stored randomness $t_u, \mathbf{\Omega}_u$ and $\{\omega_{u,j}\}_{j \in [\rho]}$, since it will not be needed to process updates to any other point in the stream. The total space required to compute the entire sketches $\mathbf{S}^1\mathbf{D}^1\mathbf{X}, \mathbf{S}^2\mathbf{D}^2\mathbf{Y}$ is $\tilde{O}(d + 1/\epsilon)$, where the $\tilde{O}(d)$ comes from storing the randomness in the blocks B_u, the Quadtree randomness, and the hash functions h_i, h_{i-1}, and $\tilde{O}(1/\epsilon)$ is the space required to store the sketches $\mathbf{S}^1\mathbf{D}^1\mathbf{X}, \mathbf{S}^2\mathbf{D}^2\mathbf{Y}$. Since after the universe reduction step we have $|\mathcal{V}_{i-1}|, |\mathcal{V}_i| = \text{poly}(s)$, our algorithm requires a total of $\text{poly}(s, d)$ bits of randomness, and can be tested in by a space $\tilde{O}(d + 1/\epsilon)$ algorithm that reads the random bits to be deradnomized in a stream. Thus, applying Nisans PRG [Nis92], it follows that the algorithm can be derandomized with a multiplicative increase of $O(\log sd)$ bits of space over the space of the tester, which completes the proof.

■

6.5.1 Proofs of Lemmas 6.8 and 6.9

We now provide the proofs of Lemmas 6.8 and 6.9. We restate the Lemmas again here for convenience.

Lemma 6.8 *Let $v^* \in \mathcal{V}_i$ be the vertex which is sampled in Figure 6.5. Then for any $v \in \mathcal{V}_i$, we have $\mathbf{Pr}[v^* = v] = (1 \pm \epsilon_0)\frac{||A_v|-|B_v||}{\Delta_i} \pm s^{-c}$, where $c > 1$ is an arbitrarily large constant.*

Proof: We first condition on the success of the count-sketch matrices $\mathbf{S}^1, \mathbf{S}^2$ on all $O(\rho)$ columns of $\mathbf{X}, \mathbf{Y}$, which by a union bound occurs with probability $1 - s^{-c}$ for any constant c by Theorem 11 and Theorem 12. The possibility of the failure of this event will only introduce an additive s^{-c} into the variation distance of the distribution of the sampler, which is safely within the guarantees of the theorem. In the following, C, C', C'' will be sufficiently large constants.

It will now suffice to bound our error in estimating the target vectors, since we can then apply Lemma 6.5. First, we show that $|\tilde{\mathbf{Z}}^1_{v,j} - (\mathbf{D}^1\mathbf{X})_{v,j}| \leq \eta\Delta_i$ with probability $1 - 1/\text{poly}(s)$ for each of the first $j \in [\rho]$ columns. By the Count-Sketch guarantee, we know that $|\tilde{\mathbf{Z}}^1_{v,j} - (\mathbf{D}^1\mathbf{X})_{v,j}| \leq \eta\|(\mathbf{D}^1\mathbf{X}_{*,j})_{-1/\eta^2}\|_2$. First, since inverse exponentials satisfies $\mathbf{Pr}[1/x > t] < 1/t$ for all $t > 1$, by Lemma 6.6 we have $\|(\mathbf{D}^1\mathbf{X}_{*,j})_{-1/\eta^2}\|_2 \leq C\eta\|(\mathbf{X}_{*,j})_{-1/(2\eta^2)}\|_1$ with probability $1 - 1/\text{poly}(s)$ for some constant C. Next, recall that $\mathbf{X}_{u,j} = \langle \lambda_u, \omega_{u,j} \rangle$, where $\omega_{u,j}$ is a vector of independent Cauchy random variables, which by 1-stability of the Cauchy distribution is itself distributed as $\omega\|\lambda_u\|_1 = \omega\Delta_i(u)$ where ω is a Cauchy random variable. Thus the column vector $\mathbf{X}_{*,j}$ is the result of scaling the column vector with coordinates equal to $\Delta_i(u)$ for each $u \in \mathcal{V}_{i-1}$ by independent Cauchy random variables, which also satisfy the tail bound $\mathbf{Pr}[1/x > t] < 1/t$ for all $t > 1$. So applying Lemma 6.6 on the non-zero rows of $\mathbf{X}$ (of which there are at most $2s$), we have

$$\|(\mathbf{X}_{*,j})_{-1/(2\eta^2)}\|_1 \leq C \log s \sum_{u \in \mathcal{V}_{i-1}} \Delta_i(u) = (C \log s) \cdot \Delta_i$$

with probability $1 - s^{-c}$. Butting these bounds together, we have $|\tilde{\mathbf{Z}}^1_{u,j} - (\mathbf{D}^1\mathbf{X})_{u,j}| \leq (C'\eta^2 \log s)\Delta_i$ for all $u \in \mathcal{V}_{i-1}$ and $j \in [\rho]$. It follows that setting $\tilde{\Delta}_i(u) = \text{median}_{j\in[\rho]}|\tilde{\mathbf{Z}}^1_{u,j}|$, the median can change by at most the error $(2C'\eta^2 \log s)\Delta_i$, so

$$\left|\tilde{\Delta}_i(u) - \frac{1}{t_u}\text{median}_{j\in[\rho]}|\mathbf{X}_{u,j}|\right| \leq (2C'\eta^2 \log s)\Delta_i$$

Thus, if we define a vector $a \in \mathbb{R}^{|\mathcal{V}_{i-1}|}$ via the coordinates $a_u = \text{median}_{j\in[\rho]}|\mathbf{X}_{u,j}|$, we obtain an entry-wise approximation $\tilde{\Delta}_i(u)$ to the scaling $\mathbf{D}^1 a$ with error at most $(2C'\eta^2 \log s)\Delta_i$. By standard arguments for p-stable variables [Ind06b], we have that $\text{median}_{j\in[\rho]}|\mathbf{X}_{u,j}| = (1 \pm \epsilon_0)\Delta_i(u)$ with probability $1 - s^{-c}$, thus $\|a\|_1 = (1 \pm \epsilon_0)\Delta_i$ after a union bound over all $2s$ non-zero coordinates. So by Lemma 6.5 applied to the starting vector a, we have that if $u^* = \arg\max_u . \tilde{\Delta}_i(u)$ then we have $\mathbf{Pr}[u^* = u] = (1 \pm O(\log s\eta^2))\frac{|a_u|}{\|a\|_1} \pm s^{-c}$. Moreover, since $|a_u| = (1 \pm \epsilon_0)|\Delta_i(u)|$ for all u, and $\eta < \epsilon_0$, it follows that $\mathbf{Pr}[u^* = u] = (1 \pm 2\epsilon_0)\frac{\Delta_i(u)}{\Delta_i} \pm s^{-c}$ as needed.

Next, we move on to the sampling of v^* given u^*. We begin by bounding the tail $\|(\mathbf{D}^2\mathbf{Y}_{*,1})_{-1/\eta^2}\|_2$. Similarly as before, we can bound this by $C\eta\sum_{v\in\mathcal{V}_i}\left(|\mathbf{Y}_{v,1}|/t_{\pi(v)}\right)$ with probability $1 - s^{-c}$ using Lemma 6.6 applied to the exponential t_v for the children. Note that $\sum_{v\in\mathcal{V}_i}\left(|\mathbf{Y}_{v,1}|/t_{\pi(v)}\right) =$

$\|b\|_1$, where b is the vector with coordinates $b_u = \Delta_i(u)/t_u$. Next, we will prove that $\|b\|_1 \leq C'' \log s(\Delta_i(u^*)/t_{u^*} + \Delta_i)$ with high probability. To see this, notice $\|b\|_1 \leq \log s \|b\|_\infty + \|b_{-\log s}\|_1$, so we can apply Lemma 6.6 to bound $\|b_{-\log s}\|_1 \leq (C \log s)\Delta_i$ with probability $1 - s^{-c}$. Since our algorithm choose u^* as the maximizer of b, we have

$$\left| \frac{\Delta_i(u^*)}{t_{u^*}} - \|b\|_\infty \right| \leq (1 + O(\epsilon_0))2C'\eta^2 \log s \Delta_i < \frac{\Delta_i}{100c \log s}$$

where c is a constant, which follows due to our count-sketch error in estimating $\mathbf{Z}$ as argued above, after setting $\eta = \Omega\left(\sqrt{\frac{\epsilon}{\log d \log^\ell s}}\right)$ with a small enough constant, where $\ell \geq 2$. Given this, it follows that $\|b\|_\infty < \Delta_i(u^*)/t_{u^*} + \Delta_i/(\log s)$, from which the bound $\|b\|_1 \leq C'' \log s(\Delta_i(u^*)/t_{u^*} + \Delta_i)$ follows. Now notice that since $\|b\|_\infty$ is the max order statistic of a set of independent exponential, by the results of Section 6.1 we have the distributional equality $\|b\|_\infty = \Delta_i/E_1$ where E_1 is an exponential random variable, so with probability $1 - s^{-c}$ we have $\|b\|_\infty > \Delta_i/(c \log s)$, thus we also have

$$\frac{\Delta_i(u^*)}{t_{u^*}} > \frac{1}{2}\|b\|_\infty > \frac{\Delta_i}{2c \log s} \tag{41}$$

Thus, plugging everything into the Count-Sketch guarantee, we obtain

$$\begin{aligned} \left| \tilde{\mathbf{Z}}^2_{v,1} - \mathbf{D}^2 \mathbf{Y}_{v,1} \right| &\leq C'' \eta^2 \log s \left(\frac{\Delta_i(u^*)}{t_{u^*}} + \Delta_i \right) \\ &\leq O\left(\eta^2 \log^2 s \left(\frac{\Delta_i(u^*)}{t_{u^*}} \right) \right) \end{aligned} \tag{42}$$

for all $v \in \mathcal{V}_i$. Recall that we sample v^* by choosing $v^* = \arg\max_{v:\pi(v)=u^*} |\tilde{\mathbf{Z}}^2_{v,1}|$. We already have a bound on the error of the estimation given by $\tilde{\mathbf{Z}}^2_{*,1}$. Thus, to apply Lemma 6.5, the only remaining piece is to notice that the original ℓ_1 norm of the vector we are sampling from is $\sum_{v:\pi(v)=u^*} \mathbf{Y}_{v,1} = \frac{1}{t_{u^*}} \sum_{v:\pi(v)=u^*} ||A_v| - |B_v|| = \frac{\Delta_i(u)}{t_u}$ after first fixing the parent exponentials t_u. Thus we can apply Lemma 6.5 with the error parameter $O(\eta^2 \log^2 s)$, which is at most ϵ_0 after setting η with a small enough constant, to obtain that $\mathbf{Pr}[v^* = v \mid u^*] = (1 \pm \epsilon_0)\frac{||A_v| - |B_v||}{\Delta_i(u^*)} \pm s^{-c}$. Noting that the randomness for which this event is determined only depends on the randomness in the second set of exponentials t_v^2, we have that

$$\begin{aligned} \mathbf{Pr}[v^* = v] &= \left((1 \pm \epsilon_0)\frac{||A_v| - |B_v||}{\Delta_i(u^*)} \pm s^{-c} \right) \left((1 \pm 2\epsilon_0)\frac{\Delta_i(u^*)}{\Delta_i} \pm s^{-c} \right) \\ &= (1 \pm 4\epsilon_0)\frac{||A_v| - |B_v||}{\Delta_i} \pm 3s^{-c} \end{aligned} \tag{43}$$

which yields the desired theorem after rescaling of c and ϵ_0 by a constant. ■

To prove the next main Lemma, Lemma 6.9, we will need a few minor technical propositions.

Proposition 6.12. *Let X be an inverse exponential random variable, and let Y be a random variable supported on $[1, \infty)$ with the property that for all $t \geq 1$:*

$$\mathbf{Pr}[Y > t] < \frac{f(t)}{t}$$

for some non-decreasing function $f : [1,\infty) \to [1,\infty)$ with $f(t) = o(t)$. Then there is a universal constant C such that $\mathbf{Pr}[XY > t] \leq C\frac{f(t)\log(t+1)}{t}$ for all $t \geq 1$.

Proof: Note that X has the probability density function $p(x) = \frac{1}{x^2}e^{-\frac{1}{x}}$. Thus, we have

$$\begin{aligned}
\mathbf{Pr}[XY > t] &\leq \int_{x=0}^{t} \mathbf{Pr}\left[Y > \frac{t}{x}\right] \frac{1}{x^2}e^{-\frac{1}{x}}dx + \int_{x=t}^{\infty} \frac{1}{x^2}e^{-\frac{1}{x}}dx \\
&\leq \int_{x=0}^{1} \mathbf{Pr}[Y > t]dx + \int_{x=1}^{t} \mathbf{Pr}\left[Y > \frac{t}{x}\right] \frac{1}{x^2}e^{-\frac{1}{x}}dx + O(\frac{1}{t}) \\
&\leq O\left(\frac{f(t)}{t}\right) + \frac{f(t)}{t}\int_{x=1}^{t} \frac{1}{x}e^{-\frac{1}{x}}dx \\
&\leq O\left(\frac{f(t)\log(t+1)}{t}\right)
\end{aligned} \tag{44}$$

where in the second line, we used that $e^{-1/x}/x^2 < 1$ for all $x > 0$. ■

Proposition 6.13. *Let $x \in \mathbb{R}^n$ be any vector with $\|x\|_1 = s$. Let X be an inverse exponential random variable, and let $Y_1, Y_2, \ldots, Y_n$ be independent random variables supported on $[1,\infty)$,with the property that $\mathbf{Pr}[Y_i > t] < f(t)/t$ for each i and all $t > 1$, where $f(t) = \log^{O(1)}((t+1)\text{poly}(s))$. Let $\lambda_1, \lambda_2, \ldots, \lambda_n$ be independent variables, where $\lambda_i \sim X_iY_i$, and X_i is an independent copy of X. Let $z \in \mathbb{R}^n$ be defined via $z_i = x_i\lambda_i$. Then we have $\|z_{-\beta}\|_1 \leq C(\log s)(\log n)f(s)\|x\|_1$ for a fixed constant C, with probabiltiy $1 - e^{-\beta/8}$.*

Proof: We have $\mathbf{Pr}[z_i > C\log sf(s)\|x\|_1] \leq \frac{x_i}{10\|x\|_1}$ by Proposition 6.12, for some sufficiently large constant C. By Chernoff bounds, with probability $1 - s^{-\beta}$ there are at most $\beta/2$ coordinates $i \in [n]$ with $z_i > C\log sf(s)\|x\|_1$. Thus if we truncate the variables λ_i into new variables λ_i' by enforcing that

$$\lambda_i' < 100\log sf(s)\frac{\|x\|_1}{x_i}$$

for all $i \in [n]$, and set $z' = \lambda_i'x_i$, the we have $\|z_{-\beta}\|_1 \leq \|z'_{-\beta/2}\|_1$. We can then apply Lemma 6.6 on the truncated variables to bound $\|z'_{-\beta/2}\|_1$, noting that for all $t \geq 1$ we have $\mathbf{Pr}[\lambda_i' > t] < \frac{\alpha}{t}$ where $\alpha = O(f(s)\log s)$, to obtain the proposition. ■

We will also need the following technical lemma. It appears as a special case of Lemma 1 in [JW19], but with the assumption that X_i's are non-negative vectors, and a tighter bound of $\|\sum_i X_i\|_1$ instead of $\sum_i \|X_i\|_1$ (note that this is only tighter if the X_i's are not non-negative). Our vectors will in fact be non-negative, but we provide a simple proof for version of the lemma where the vectors may have negative coordinates.

Proposition 6.14 (Special case of Lemma 1[JW19])**.** *Let $Z \in \mathbb{R}^d$ be a vector of i.i.d. Cauchy random variables. Let $X_1, \ldots, X_k \in \mathbb{R}^d$ be fixed vectors. Then there is a fixed constant C, such that for any $t \geq 1$, we have*

$$\mathbf{Pr}\left[\sum_{i=1}^{k} |\langle Z, X_i\rangle| \geq C\log(tk)t \cdot \sum_{i=1}^{k} \|X_i\|_1\right] \leq \frac{1}{t}$$

Proof: The quantity $\langle \omega, p_i \rangle$ is distributed as $\alpha_i \|p_i\|_1$ where the α_i's are *non-independent* Cauchy random variables. Let $\mathcal{E}_i$ be the event that $|\alpha_i| \leq 1/\delta$, which occurs with probability $1-\delta$ by the tails of Cauchy variables. Let $\mathcal{E} = \cap_i \mathcal{E}_i$, and note $\mathbf{Pr}[\mathcal{E}] > 1 - O(k\delta)$ by a union bound. We have

$$\mathbf{E}[|a_i| \mid \mathcal{E}_i] \leq 3 \int_{x=0}^{1/\delta} \frac{x}{\pi(1+x^2)} \leq \log(1/\delta)/2$$

It follows that $\mathbf{E}[|a_i| \mid \mathcal{E}] \leq 2\mathbf{E}[|a_i| \mid \mathcal{E}_i]$ since $\mathbf{Pr}[\mathcal{E}|\mathcal{E}_i] > 1 - 2$, and so:

$$\mathbf{E}\left[\sum_{i=1}^{k} |\alpha_i| \|p_i\|_1 \mid \mathcal{E}\right] \leq C \log(1/\delta) \sum_{i=1}^{k} \|p_i\|_1$$

Setting $\delta < \frac{1}{2tk}$, by Markov's inequality.

$$\mathbf{Pr}\left[\sum_{i=1}^{k} |\langle Z, X_i \rangle| \geq 2C \log(2tk) t \cdot \sum_{i=1}^{k} \|X_i\|_1\right] \leq \frac{1}{2t} + \mathbf{Pr}[\mathcal{E}] < \frac{1}{t}$$

Which yields the proposition after rescaling C by constant. ■

Lemma 6.9 *Fix $\eta = \Theta\left(\sqrt{\frac{\epsilon}{\log d \log^{\ell} s}}\right), \epsilon_0 = \Theta(1/\log s)$, and let (u^*, v^*) be the samples accepted by Figure 6.5. Then assuming that $\frac{2s}{\Delta_i} < \frac{\log s \log d}{\epsilon 2^i}$, and moreover that $\frac{|A_{u^*}|+|B_{u^*}|}{\Delta_i(u^*)} \leq \frac{\log s \log d}{\epsilon \nu 2^i}$, where $\nu = \Theta(1/\log s)$, then with probability $1 - 1/\mathrm{poly}(s)$, we have*

$$\frac{1}{\mathrm{median}(|\mathcal{C}|)} \cdot \mathrm{median}_{j \in [\rho]} \left| \frac{\tilde{\mathbf{Z}}^1_{u^*,\rho+j+1}}{\tilde{\mathbf{Z}}^1_{u^*,\rho+1}} - \frac{\tilde{\mathbf{Z}}^2_{v^*,j+2}}{\tilde{\mathbf{Z}}^2_{v^*,2}} \right| = (1 \pm \epsilon_0)\|c_{u^*} - c_{v^*}\|_1 \pm \epsilon_1 \frac{d}{2^i}$$

where ϵ_1 is an arbitrarily small constant.

Proof: First suppose that our output was actually $\mathrm{median}_{j \in [\rho]} \left| \frac{\mathbf{Z}^1_{u^*,\rho+j+1}}{\mathbf{Z}^1_{u^*,\rho+1}} - \frac{\mathbf{Z}^2_{v^*,j+2}}{\mathbf{Z}^2_{v^*,2}} \right|$, and let us see that this would be a good approximation. Noticed that since each row u of $\mathbf{Z}^1$ is scaled by the same value $1/t_u$, and similarly with $\mathbf{Z}^2$, this is the same as the quantity $\mathrm{median}_{j \in [\rho]} \left| \frac{\mathbf{X}_{u^*,\rho+j+1}}{\mathbf{X}_{u^*,\rho+1}} - \frac{\mathbf{Y}_{v^*,j+2}}{\mathbf{Y}_{v^*,2}} \right|$. Plugging in definitions, this is just

$$\begin{aligned} &= \mathrm{median}_{j \in [\rho]} \left| \frac{\left(\Omega_u \left(\sum_{p \in A_u \cup B_u} p\right)\right)_j}{||A_u| + |B_u||} - \frac{\left(\Omega_u \left(\sum_{p \in A_u \cup B_u} p\right)\right)_j}{||A_v| + |B_v||} \right| \\ &= \mathrm{median}_{j \in [\rho]} \left| (\Omega_u (c_u - c_v))_j \right| \end{aligned} \tag{45}$$

By standard concentration for medians of p-stables [Ind06b], it follows that

$$\frac{\mathrm{median}_{j \in [\rho]} \left| (\Omega_u (c_u - c_v))_j \right|}{\mathrm{median}(|\mathcal{D}_1|)} = (1 \pm \epsilon_0)\|c_u - c_v\|_1 \tag{46}$$

with probability $1 - s^{-c}$.

Claim 6.15. *To prove the Lemma, it suffices to prove that both*

$$\left|\frac{\tilde{\mathbf{Z}}^1_{u^*,\rho+j+1}}{\tilde{\mathbf{Z}}^1_{u^*,\rho+1}} - \frac{\mathbf{Z}^1_{u^*,\rho+j+1}}{\mathbf{Z}^1_{u^*,\rho+1}}\right| \leq \epsilon_1 \frac{d}{2^{i+2}}, \qquad \left|\frac{\tilde{\mathbf{Z}}^2_{v^*,j+2}}{\tilde{\mathbf{Z}}^2_{v^*,2}} - \frac{\mathbf{Z}^2_{v^*,j+2}}{\mathbf{Z}^2_{v^*,2}}\right| \leq \epsilon_1 \frac{d}{2^{i+2}} \tag{47}$$

hold with probability at least $1 - 1/\zeta$ *independently for each* $j \in [\rho]$*, where* $\zeta = \Theta(1/\epsilon_0)$ *with a large enough constant.*

Proof: Let

$$\theta_j = \frac{\left|(\Omega_u(c_u - c_v))_j\right|}{\text{median}(|\mathcal{C}|)}, \quad \widehat{\theta_j} = \frac{\left|\frac{\tilde{\mathbf{Z}}^1_{u^*,\rho+j+1}}{\tilde{\mathbf{Z}}^1_{u^*,\rho+1}} - \frac{\tilde{\mathbf{Z}}^2_{v^*,j+2}}{\tilde{\mathbf{Z}}^2_{v^*,2}}\right|}{\text{median}(|\mathcal{C}|)}.$$

By the assumption and triangle inequality, we have that $|\widehat{\theta_j} - \theta_j| < \epsilon_2 d/2^{i+1}$, where $\epsilon_2 = \frac{\epsilon_1}{\text{median}(|\mathcal{D}_1|)}$ is within a fixed constant of ϵ_1, with probability at least $1 - \zeta^{-1}$. Since there are ρ repetitions, it follows by Chernoff bounds that there will be at most $2\rho/\zeta$ values of $j \in [\rho]$ such that this approximation does not hold, with probability at least $1 - 1/\text{poly}(s)$. Let $W \subset [\rho]$ be the subset of indices where this guarantee fails, where $|W| \leq 2\rho/\zeta$. Let $\theta'_j = \theta_j$ for $j \notin W$, and let $\theta'_j = \widehat{\theta_j}$ for $j \in W$. Then we have $|\widehat{\theta_j} - \theta'_j| < \epsilon_2 d/2^{i+1}$ for all $j \in [\rho]$. It follows that $|\text{median}_{j\in[\rho]}\widehat{\theta_j} - \text{median}_{j\in[\rho]}\theta'_j| \leq \epsilon_2 d/2^i$. Thus, it will suffice to show that $\text{median}_{j\in[\rho]}\theta'_j$ is a good approximation of $\|c_u - c_v\|_1$.

To see this, first note by Lemma 2 of [Ind06b], for any $\phi > 0$ smaller than a constant, if C is Cauchy we have that $\mathbf{Pr}[|C| < (1-\phi)\cdot\text{median}(|\mathcal{C}|)] < 1/2 - \phi/4$, and similarly $\mathbf{Pr}[|C| > (1+\phi)\cdot\text{median}(|\mathcal{D}_1|)] < 1/2 + \phi/4$. Then by Chernoff bounds, the number j for which θ_j is less than $(1-\epsilon_0)\|c_u - c_v\|_1$ is at most $(1-\epsilon_0/4)\rho/2$, and similarly the number of j for which this value is at least $(1+\epsilon_0)\|c_u - c_v\|_1$ is at most $(1-\epsilon_0/4)\rho/2$, both with probability $1 - s^{-c}$. Now $\theta'_1, \ldots, \theta'_\rho$ is the result of arbitrarily corrupting the value of an arbitrary subset of $2\rho/\zeta < \epsilon\rho/10$ of the values in $\theta_1, \ldots, \theta_\rho$. After any such corruption, it follows that there must still be at most $\rho/2 - \epsilon_0\rho/8 + \epsilon_0\rho/10 < \rho/2 - \epsilon_0\rho/40$ indices j with $\theta'_j < (1-\epsilon_0)\|c_u - c_v\|_1$, and at most $\rho/2 - \epsilon_0\rho/40$ indices j with $\theta'_j > (1+\epsilon_0)\|c_u - c_v\|_1$. It follows that $\text{median}_{j\in[\rho]}\theta'_j = (1\pm\epsilon_0)\|c_u - c_v\|_1$, thus $\text{median}_{j\in[\rho]}\widehat{\theta_j} = (1\pm\epsilon_0)\|c_u - c_v\|_1 + \epsilon_2 d/2^i$, which is the desired result after scaling ϵ_2 down by a constant $1/\text{median}(|\mathcal{D}_1|)$.

■

We now prove that Equation 47 holds with probability at least $1 - 1/\zeta$.We show that it holds for each of the two terms with probability at least $1 - 1/(2/\zeta)$, and the result will then follow by a union bound. In what follows, we let C, C', C'' be sufficiently large constants. We begin with the first term. First, note that by the count-sketch guarantee, we have $|\tilde{\mathbf{Z}}^1_{u^*,\rho+1} - \mathbf{Z}^1_{u^*,\rho+1}| \leq \eta\|(\mathbf{Z}_{*,\rho+1})^1_{-1/\eta^2}\|_2$. Applying Lemma 6.6, we have $\|(\mathbf{Z}_{*,\rho+1})_{-1/\eta^2}\|_2 \leq C\eta\|(\mathbf{X}_{*,\rho+1})_{-1/(2\eta^2)}\|_1 \leq 2C\eta s$, where the last inequality uses the fact that each $\mathbf{X}_{u,\rho+1} = |A_u| + |B_u|$ so $\|\mathbf{X}_{*,\rho+1}\|_1 = 2s$. Thus $|\tilde{\mathbf{Z}}^1_{u,\rho+1} - \mathbf{Z}^1_{u,\rho+1}| \leq 2C\eta^2 s$. Moreover, by Equation 41 in the proof of Lemma 6.8, we have that $\Delta_i < \frac{10c\log s\Delta_i(u^*)}{t_{u^*}}$ with probability at least $1 - s^{-c}$, which recall followed from the observation that our error from count-sketch on the first column of $\mathbf{Z}^1$ was at most $\tilde{O}(\eta^2\Delta_i)$ and that the maximum value of $\Delta_i(u)/t_u$ for $u \in \mathcal{V}_{i-1}$ is distributed like Δ_i/E_1 where E_1 is an exponential random variable, and

then we conditioned on the event $E_1 < 5c \log s$. Since $|A_{u^*}| + |B_{u^*}| \geq \Delta_i(u^*)$, it follows that $\mathbf{Z}^1_{u^*,\rho+1} > \Delta_i/(2c \log s)$. Using that $\frac{2s}{\Delta_i} < \frac{\log s \log d}{\epsilon 2^i}$, we have $2C\eta^2 s < \frac{\epsilon s}{\log d \log^\ell s} < \frac{\Delta_i}{2^{i+1} \log^{\ell-1} s}$, so

$$\tilde{\mathbf{Z}}^1_{u^*,\rho+1} = \frac{1}{t_{u^*}}(1 \pm \epsilon_0^2 2^{-i})\left(|A_{u^*}| + |B_{u^*}|\right)$$

Next, we consider $\tilde{\mathbf{Z}}^1_{u^*,\rho+j+1}$ for $j = 1, 2, \ldots, \rho$. Applying the same argument as above, we have $|\tilde{\mathbf{Z}}^1_{u^*,\rho+j+1} - \mathbf{Z}^1_{u^*,\rho+j+1}| \leq C\eta^2 \|(\mathbf{X}_{*,\rho+1})_{-1/(2\eta^2)}\|_1$. Note by 1-stability that $\|(\mathbf{X}_{*,\rho+1})_{-1/(2\eta^2)}\|_1$ is distributed as $\sum_{u \in \mathcal{V}_{i-1}} \alpha_{u,j} \| \sum_{p \in A_u \cup B_u} p\|_1$ where $\alpha_{u,j}$ are i.i.d. Cauchy random variables. Thus we can apply Lemma 6.6 to obtain $\|(\mathbf{X}_{*,\rho+1})_{-1/(2\eta^2)}\|_1 \leq C \log s(2sd)$ with probability $1 - 1/\text{poly}(s)$, where we used that that $\sum_{u \in \mathcal{V}_{i-1}} \| \sum_{p \in A_u \cup B_u} p\|_1 \leq 2sd$. Taken together, we have

$$|\tilde{\mathbf{Z}}^1_{u^*,\rho+j+1} - \mathbf{Z}^1_{u^*,\rho+j+1}| \leq C'\eta^2 \log s \cdot (sd)$$

for all $j \in [\rho]$ with probability $1 - 1/\text{poly}(s)$ after a union bound. Now let $\mathcal{Q}^1_j$ denote the event that $|(\Omega_u c_u)_j| \leq Cd\zeta$, where $\zeta = \Theta(1/\epsilon_0)$, (where C is taken as a large enough constant) Because $(\Omega_u c_u)_j$ is distributed as $\omega \|c_u\|_1$, where ω is a Cauchy random variable, and because $\|c_u\|_1 \leq d$, we have $\mathbf{Pr}[\mathcal{Q}^1_j] > 1 - 1/(2\zeta)$ independently for separate $j \in [\rho]$. Conditioned on this, we have

$$\begin{aligned}
\frac{\tilde{\mathbf{Z}}^1_{u^*,\rho+j+1}}{\tilde{\mathbf{Z}}^1_{u^*,\rho+1}} &= (1 \pm 2\epsilon_1 2^{-i}) t_{u^*} \cdot \frac{\tilde{\mathbf{Z}}^1_{u^*,\rho+j+1}}{|A_{u^*}| + |B_{u^*}|} \\
&= (1 \pm 2\epsilon_0^2 2^{-i}) t_{u^*} \cdot \frac{\mathbf{Z}^1_{u^*,\rho+j+1} \pm C'\eta^2 \log s \cdot (sd)}{|A_{u^*}| + |B_{u^*}|} \\
&= (1 \pm 2\epsilon_0^2 2^{-i})(\Omega_u c_u)_j \pm 2t_{u^*} \frac{C'\eta^2 \log s \cdot (sd)}{|A_{u^*}| + |B_{u^*}|} \\
&= (\Omega_u c_u)_j \pm C'\left(\epsilon_0^2 2^{-i} |(\Omega_u c_u)_j| + \eta^2 \log^2 s \cdot \frac{sd}{\Delta_i}\right) \\
&= (\Omega_u c_u)_j \pm \left(\epsilon_1 \frac{d}{2^{i+2}} + C'\eta^2 \log^3 s \cdot \frac{d \log d}{\epsilon 2^i}\right) \\
&= (\Omega_u c_u)_j \pm \epsilon_1 \frac{d}{2^{i+1}}
\end{aligned} \tag{48}$$

Where we used the the fact that $\frac{|A_{u^*}|+|B_{u^*}|}{t_{u^*}} \geq \frac{\Delta_i(u^*)}{t_{u^*}} \geq \Delta_i/(2c \log s)$, and the assumptions $\frac{2s}{\Delta_i} < \frac{\log s \log d}{\epsilon 2^i}$ of the Lemma. Noting that $(\Omega_{u^*} c_{u^*})_j = \frac{\mathbf{Z}^1_{u^*,\rho+j+1}}{\mathbf{Z}^1_{u^*,\rho+1}}$, we have proven the first inequality with the desired probability.

To prove the second inequality from Claim 6.15, we first analyze $|\tilde{\mathbf{Z}}^2_{v^*,2} - \mathbf{Z}^2_{v^*,2}|$. Let $U_0 \subset \mathcal{V}_{i-1}$ be the top $1/\epsilon_0$ coordinates in magnitude of the vector $\mathbf{Z}^1_{*,\rho+1}$, which recall has coordinates $\mathbf{Z}^1_{u,\rho+1} = \sum_{v \in \mathcal{V}_i} \frac{1}{t_{\pi(v)}}(|A_v| + |B_v|)$. Applying the stronger nested count-sketch guarantee of Theorem 12, we have:

$$|\tilde{\mathbf{Z}}^2_{v^*,2} - \mathbf{Z}^2_{v^*,2}| \leq \|(\mathbf{Z}^2_{*,2})_{-(U_0 \setminus u^*, \eta^{-2})}\|_2$$

with probability $1 - s^{-c}$, where recall $(\mathbf{Z}^2_{*,2})_{-(U_0 \setminus u^*, \eta^{-2})}$ is the notation from Definition 6.7, and is defined as the result of first zero-ing out all rows v of $\mathbf{Z}^2_{*,2}$ with parents $\pi(v) \in U_0 \setminus u^*$, and then

removing the top η^{-2} largest of the remaining coordinates. We can now apply Lemma 6.6 on the exponential scalings t_v for the children to obtain

$$\begin{aligned}\|(\mathbf{Z}^2_{*,2})_{-(U_0\setminus u^*,\eta^{-2})}\|_2 &\leq C\eta\left(\sum_{u\in(\mathcal{V}_{i-1}\setminus U_0)\cup u^*}\frac{1}{t_u}\sum_{v:\pi(v)=u}|A_v|+|B_v|\right)\\ &= C\eta\left(\|\left(\mathbf{Z}^1_{*,\rho+1}\right)_{-1/\epsilon_0}\|_1+\frac{1}{t_{u^*}}(|A_{u^*}|+|B_{u^*}|)\right)\end{aligned} \tag{49}$$

We can apply then Lemma 6.6 to obtain $\|\left(\mathbf{Z}^1_{*,\rho+1}\right)_{-1/\epsilon_0}\|_1 \leq C\log s\|\mathbf{X}^1_{*,\rho+1}\|_1 = (2C\log s)\cdot s$ with probability $1-s^{-c}$, where we used that $\|\mathbf{X}^1_{*,\rho+1}\|_1$ is simply the number of points in $A\cup B$. This yields

$$\begin{aligned}\left|\tilde{\mathbf{Z}}^2_{v^*,2}-\mathbf{Z}^2_{v^*,2}\right| &\leq C\eta^2\left(\log s\cdot s+\frac{1}{t_{u^*}}(|A_{u^*}|+|B_{u^*}|)\right)\\ &\leq C\eta^2\left(\cdot\frac{\Delta_i\log^2 s\log d}{\epsilon 2^i}+\frac{1}{t_{u^*}}\cdot\frac{\Delta_i(u^*)\log s\log d}{\epsilon\nu 2^i}\right)\\ &\leq C'\eta^2\left(\frac{c}{t_{u^*}}\cdot\frac{\Delta_i(u^*)\log^3 s\log d}{\epsilon\nu 2^i}\right)\\ &\leq \epsilon_0^2\left(\frac{\Delta_i(u^*)}{t_{u^*}2^i}\cdot\right)\end{aligned} \tag{50}$$

Where in the second inequality we used the two assumptions $\frac{2s}{\Delta_i}<\frac{\log s\log d}{\epsilon 2^i}$ and $\frac{|A_{u^*}|+|B_{u^*}|}{\Delta_i(u^*)}\leq\frac{\log s\log d}{\epsilon\nu 2^i}$ of the Lemma, and in the third inequality we again used the earlier fact from Equation 41 in the proof of Lemma 6.8 that $\Delta_i<\frac{10c\log s\Delta_i(u^*)}{t_{u^*}}$ with probability at least $1-s^{-c}$. Similarly, we know that $\frac{||A_{v^*}|-|B_{v^*}||}{t_{u^*}t_{v^*}}\geq\frac{\Delta_i(u^*)}{2ct_{u^*}\log s}$ with probability $1-s^{-c}$, applying the lower bound on the top scaled coordinate as in the proof of the last Lemma. Thus $\mathbf{Z}^2_{v^*,2}=\frac{||A_{v^*}|+|B_{v^*}||}{t_{u^*}t_{v^*}}\geq\frac{||A_{v^*}|-|B_{v^*}||}{t_{u^*}t_{v^*}}\geq\frac{\Delta_i(u^*)}{2ct_{u^*}\log s}$, so using our bound on the error and $\epsilon_0=\Theta(1/\log s)$, we have

$$\tilde{\mathbf{Z}}^2_{v^*,2}=\frac{1}{t_{u^*}t_{v*}}(1\pm\epsilon_0 2^{-i})\left(|A_{v^*}|+|B_{v^*}|\right)$$

Next, we must consider the estimates $\tilde{\mathbf{Z}}^2_{v^*,j+2}$. Let $\mathbf{M}^j\in\mathbb{R}^{|\mathcal{V}_{i-1}|}$ be the vector given by $\mathbf{M}^j_u=\frac{1}{t_u}\sum_{v\,:\,\pi(v)=u}|\mathbf{Y}_{v,j+2}|$ for $u\in\mathcal{V}_{i-1}$ and $j\in[\rho]$. By Lemma 6.14, we can write

$$\mathbf{M}^j_u=\alpha_u\sum_{v\,:\,\pi(v)=u}C\left\|\sum_{p\in A_v\cup B_v}p\right\|_1$$

where C is a constant, $\alpha_u=\frac{1}{t_u}\beta_u$, and $\{\beta_u\}$ are positive independent (non-identical) variables with tails $\mathbf{Pr}[\beta_i>t]\leq\frac{\log(ts)}{t}$. Now define $U\subset\mathcal{V}_{i-1}$ to be the set of $1/\epsilon_0$ largest coordinates u of $\mathbf{M}$. We can then apply Theorem 12 on U. This yields

$$|\tilde{\mathbf{Z}}^2_{v^*,j+2}-\mathbf{Z}^2_{v^*,j+2}|\leq\eta\|(\mathbf{D}^2\mathbf{Y}_{*,j+2})_{-(U\setminus u^*,\eta^{-2})}\|_2$$

for any $j \in [\rho]$ with high probability. Recall that $(\mathbf{D}^2\mathbf{Y}_{*,j+2})_{-(U\setminus u^*,\eta^{-2})}$ first zeros out all rows corresponding to children of $u \in U \setminus \{u^*\}$, and then removes the top $1/\eta^2$ remaining coordinates. Using this fact, we can apply Lemma 6.6, to obtain

$$|\tilde{\mathbf{Z}}^2_{v^*,j+2} - \mathbf{Z}^2_{v^*,j+2}| \leq C\eta^2 \|(\mathbf{D}^2\mathbf{Y}_{*,j+2})_{-(U\setminus u^*,0)}\|_1 = C\eta^2 \left(\|\mathbf{M}^j_{-1/\epsilon_0}\|_1 + |\mathbf{M}^j_{u^*}|\right)$$

Now notice that $\mathbf{M}^j$ has coordinates $\mathbf{M}^j_u = \alpha_u \sum_{v\,:\,\pi(v)=u} C\|\sum_{p\in A_v\cup B_v} p\|_1$, where α_u are independent, so we can apply Proposition 6.13 to the at most $2s$ non-zero rows of $\mathbf{M}^j$ using the function $f(t) \leq \log(ts)$, to obtain

$$\|\mathbf{M}^j_{-1/\epsilon_0}\|_1 \leq C\log^3(s)\left(\sum_{v\in\mathcal{V}_i} \|\sum_{p\in A_v\cup B_v} p\|_1\right) \leq C_0\log^3(s)sd$$

for some constant C_0 with probability $1 - s^{-c}$. Now let $\mathcal{Q}^2_j$ be the event that, for the random variables β defined by the vector $\mathbf{M}^j$, we have $\beta_{u^*} < C\zeta\log(s\zeta)$ for some large enough constant C, where recall $\zeta = \Theta(1/\epsilon_0)$ is chosen with a large enough constant as earlier. By Lemma 6.14, we have $\mathbf{Pr}[\mathcal{Q}^2_j] < 1/(4\zeta)$, and note that $\mathcal{Q}^2_j$ is only a function in the randomness of the j-th rows of the sketches Ω_u for $u \in \mathcal{V}_{i-1}$. In particular, the events $\mathcal{Q}^2_j$ are independent for separate $j \in [\rho]$. Now let $\mathcal{Q}^3_j$ be the event that $|(\Omega_u c_v)_j| \leq Cd\zeta$, which, as in the earlier case of $\mathcal{Q}^1_j$, holds with probability at least $1 - 1/(4\zeta)$. Letting $\mathcal{Q}^4_j = \mathcal{Q}^2_j \cup \mathcal{Q}^3_j$, we have $\mathbf{Pr}[\mathcal{Q}^4_j] > 1 - 1/(2\zeta)$ by a union bound. The event $\mathcal{Q}^4_j$ will then be the desired event, depending only on the j-th row of the sketch Ω_u, which holds with probability at least $1 - 1/(2\zeta)$. Conditioned on $\mathcal{Q}^4_j$., we have $|\mathbf{M}^j_{u^*}| \leq \frac{1}{t_{u^*}}\zeta 2d||A_{u^*}| + |B_{u^*}||$, and:

$$\begin{aligned}
\frac{\tilde{\mathbf{Z}}^2_{v^*,j+2}}{\tilde{\mathbf{Z}}^2_{v^*,2}} &= (1 \pm 2\epsilon_0 2^{-i})t_{u^*}t_{v^*} \cdot \frac{\tilde{\mathbf{Z}}^2_{v^*,j+2}}{|A_{v^*}| + |B_{v^*}|} \\
&= (1 \pm 2\epsilon_0 2^{-i})\frac{\mathbf{Z}^2_{v*,\rho+j+1}}{\mathbf{Z}^2_{v^*,2}} \pm t_{u^*}t_{v^*}\eta^2 C' \frac{(\log^3 s)s + \frac{1}{t_{u^*}}\zeta 2d||A_{u^*}| + |B_{u^*}||}{|A_{v^*}| + |B_{v^*}|} \\
&= (1 \pm 2\epsilon_0 2^{-i})\frac{\mathbf{Z}^2_{v*,\rho+j+1}}{\mathbf{Z}^2_{v^*,2}} \pm \eta^2 C' \left(t_{u^*}t_{v^*}\frac{(\log^3 s)s}{|A_{v^*}| + |B_{v^*}|} + \frac{t_{v^*}\zeta 2d||A_{u^*}| + |B_{u^*}||}{|A_{v^*}| + |B_{v^*}|}\right) \\
&= (1 \pm 2\epsilon_0 2^{-i})\frac{\mathbf{Z}^2_{v*,\rho+j+1}}{\mathbf{Z}^2_{v^*,2}} \pm \eta^2 C'' \left(\frac{(\log^5 s)s}{\Delta_i} + \frac{(\log s)\zeta d||A_{u^*}| + |B_{u^*}||}{\Delta_i(u^*)}\right) \\
&= (1 \pm 2\epsilon_0 2^{-i})\frac{\mathbf{Z}^2_{v*,\rho+j+1}}{\mathbf{Z}^2_{v^*,2}} \pm \eta^2 C'' \left(\frac{\log^6 s \cdot (\log d)d}{\epsilon 2^i} + \frac{\zeta(\log s)^2(\log d)\cdot d}{\epsilon\nu 2^i}\right) \\
&= \frac{\mathbf{Z}^2_{v*,\rho+j+1}}{\mathbf{Z}^2_{v^*,2}} \pm \left(2\epsilon_0 2^{-i}|(\Omega_u c_v)_j| + \epsilon_1\frac{d}{2^{i+2}}\right) \\
&= \frac{\mathbf{Z}^2_{v*,\rho+j+1}}{\mathbf{Z}^2_{v^*,2}} \pm \left(\epsilon_1\frac{d}{2^{i+2}} + \epsilon_1\frac{d}{2^{i+2}}\right) \\
&= \frac{\mathbf{Z}^2_{v*,\rho+j+1}}{\mathbf{Z}^2_{v^*,2}} \pm \epsilon_1\frac{d}{2^{i+1}}
\end{aligned} \tag{51}$$

with probability $1 - s^{-c}$, where C', C'' are constants, and where we used several facts including the bound on η. Chiefly, we used that $\frac{t_{v^*}}{|A_{v^*}|+|B_{v^*}|} \leq \frac{t_{v^*}}{||A_{v^*}|-|B_{v^*}||} \leq O(\frac{\log s}{\Delta_i(u^*)})$ with probability $1 - s^{-c}$,

using that the maximum value of $\frac{t_v}{||A_v|-|B_v||}$ over children v of u^* will be at least this large, and the error from count-sketch in finding v^* is more than a constant factor smaller than this value (as in the proof of the last lemma, and used earlier in this lemma). The same fact is be applied again to show $\frac{t_{u^*}}{\Delta_i(u^*)} < O(\frac{\log s}{\Delta_i})$ with the same probability. We also used the assumptions that $\frac{2s}{\Delta_i} < \frac{\log s \log d}{\epsilon 2^i}$, and moreover that $\frac{|A_{u^*}|+|B_{u^*}|}{\Delta_i(u^*)} \leq \frac{\log s \log d}{\epsilon \nu 2^i}$. Finally, we used the bound on η^2 by definition, setting $\ell = 6$ as the exponential of $\log(s)$ in η^2. Thus, we have demonstrates Equation 47 holds with probability at least $1 - 1/\zeta$ independently for each $j \in [\rho]$, which completes the proof of the lemma. ■

Acknowledgments

We would like to thank David Woodruff, Ilya Razenshteyn, and Aleksandar Nikolov for illuminating discussions relating to this work.

A Analysis of ComputeEMD via Tree Embeddings

We sketch how the natural analysis of ComputeEMD(A, B, d) yields a $O(\min\{\log s, \log d\} \log s)$-approximation. The analysis proceeds via the method of randomized tree embeddings, and immediately gives a (randomized) embedding $\boldsymbol{f} \colon (\{0,1\}^d)^s \to \ell_1$ satisfying

$$\mathrm{EMD}(A, B) \leq \|\boldsymbol{f}(A) - \boldsymbol{f}(B)\|_1 \leq O(\min\{\log s, \log d\} \log s)\, \mathrm{EMD}(A, B),$$

with probability 0.9 over the draw of $\boldsymbol{f}$. Specifically, let $A, B \subset \{0,1\}^d$ be two multi-sets of size s, and let $M^* \subset A \times B$ be the matching such that

$$\mathrm{EMD}(A, B) = \sum_{(a,b) \in M^*} \|a - b\|_1.$$

Consider the execution tree $\mathbf{T}_0$ in described the beginning of Section 3, where we execute the algorithm for at most $O(d)$ rounds of recursion. We assign weights to the edges, where an edge connecting a node at depth i and $i+1$ is given weight $d/(i+1)^2$, which defines a tree metric $(A \cup B, d_{\mathbf{T}_0})$ given by the sum of weights over the paths connecting two points in the tree.

The following claim is a simple observation, which follows from a greedy construction of the matching over a tree. See Figure 6.

Claim A.1 (Greedy Bottom-Up Approach is Optimal for a Tree)**.** *Let $\mathbf{M} \subset A \times B$ be the matching that the execution tree $\mathbf{T}_0$ outputs, then*

$$\sum_{(a,b) \in \mathbf{M}} d_{\mathbf{T}_0}(a, b) \leq \sum_{(a,b) \in M^*} d_{\mathbf{T}_0}(a, b).$$

Lemma A.2. *With probability 0.9 over the draw of $\mathbf{T}_0$,*

$$\sum_{(a,b) \in M^*} d_{\mathbf{T}_0}(a, b) \leq O(\min\{\log s, \log d\}) \cdot \mathrm{EMD}(A, B),$$

and every $a \in A$ and $b \in B$ satisfies $d_{\mathbf{T}_0}(a, b) \geq \Omega\left(\|a - b\|_1 / \log s\right)$.

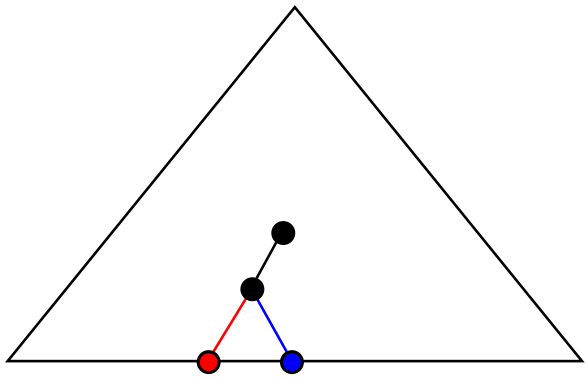

Figure 6: Proof sketch of Claim A.1. The matching $\mathbf{M}$ satisfies that for any node v in the tree, the pairs $(a,b) \in \mathbf{M}$ within the subtree rooted at v forms a maximal matching of nodes in A and B within the subtree rooted at v. In order to see why this is optimal for a tree with positive edge weights, suppose the red point is in A and the blue point in B. These meet at the (lower) black node v, but if they both remain unmatched at the upper-black node u, then both must pay the black edge.

Proof: For $(a,b) \in A \times B$ with $a \neq b$, let

$$i_{\min}(a,b) = \left\lfloor \frac{d}{\|a-b\|_1 \cdot s^3} \right\rfloor \qquad i_{\max}(a,b) = \max\left\{ \left\lceil \frac{10d \log s}{\|a-b\|_1} \right\rceil, d \right\}.$$

Then, we consider the random variable

$$\mathbf{D}(a,b) \stackrel{\text{def}}{=} 2 \sum_{i=i_{\min}(a,b)}^{i_{\max}(a,b)} \mathbf{1}\left\{(a,b) \text{ first split in } \mathbf{T}_0 \text{ at depth } i\right\} \sum_{j \geq i} \frac{d}{(i+1)^2}\cdot,$$

and notice that this is equal to $d_{\mathbf{T}_0}(a,b)$ whenever (a,b) are first split between depth $i_{\min}(a,b)$ and $i_{\max}(a,b)$. Then, we have

$$\begin{aligned}\mathop{\mathbf{E}}_{\mathbf{T}_0}\left[\mathbf{D}(a,b)\right] &\leq 2 \sum_{i=i_{\min}(a,b)}^{i_{\max}(a,b)} \mathop{\mathbf{Pr}}_{\mathbf{T}_0}\left[(a,b) \text{ first split at depth } i\right] \sum_{j \geq i} \frac{d}{(j+1)^2} \lesssim \sum_{i=i_{\min}(a,b)}^{i_{\max}(a,b)} \frac{\|a-b\|_1}{d} \cdot \frac{d}{i+1} \\ &= \|a-b\|_1 \cdot O\left(\log\left(\frac{i_{\max}(a,b)}{i_{\min}(a,b)+1}\right)\right) = \|a-b\|_1 \cdot O\left(\min\{\log s, \log d\}\right).\end{aligned}$$

Furthermore, the probability that there exists some $(a,b) \in A \times B$ such that (a,b) are not split between levels $i_{\min}(a,b)$ and $i_{\max}(a,b)$ is at most

$$\begin{aligned}&\sum_{a \in A} \sum_{\substack{b \in B \\ a \neq b}} \mathop{\mathbf{Pr}}_{\mathbf{T}_0}\left[(a,b) \text{ first split outside depths } i_{\min}(a,b) \text{ and } i_{\max}(a,b)\right] \\ &\qquad \leq \sum_{a \in A} \sum_{\substack{b \in B \\ a \neq b}} \left(\frac{i_{\min}(a,b) \cdot \|a-b\|_1}{ds^3} + \left(1 - \frac{\|a-b\|_1}{d}\right)^{i_{\max}(a,b)} \right) \leq \frac{2}{s}.\end{aligned}$$

Hence, with probability $1 - 2/s$, every $a \in A$ and $b \in B$, with $a \neq b$, satisfies $d_{\mathbf{T}_0}(a,b) \gtrsim d/i_{\max}(a,b) = \Omega(\|a-b\|_1/\log s)$, and

$$\sum_{(a,b)\in M^*} d_{\mathbf{T}_0}(a,b) = \sum_{(a,b)\in M^*} \mathbf{D}(a,b).$$

By Markov's inequality,

$$\sum_{(a,b)\in M^*} \mathbf{D}(a,b) \leq 100 \cdot O(\min\{\log s, \log d\}) \sum_{(a,b)\in M^*} \|a-b\|_1 = O(\min\{\log s, \log d\})\,\mathrm{EMD}(A,B),$$

with probability $99/100$, so that with probability $99/100 - 2/s$, we obtain the desired lemma. ■

In order to see that Claim A.1 and Lemma A.2, notice that with probability 0.9, we have the following string of inequalities:

$$\begin{aligned}\sum_{(a,b)\in\mathbf{M}} \|a-b\|_1 &\lesssim \log s \sum_{(a,b)\in\mathbf{M}} d_{\mathbf{T}_0}(a,b) \leq \log s \sum_{(a,b)\in M^*} d_{\mathbf{T}_0}(a,b)\\ &\lesssim O(\min\{\log s, \log d\}\log s)\,\mathrm{EMD}(A,B),\end{aligned}$$

where the first and last inequality follow from Lemma A.2, and the middle inequality is Claim A.1.

B Tightness of ComputeEMD

We show that we cannot improve the approximation factor of ComputeEMD beyond $O(\log s)$.

Lemma B.1. *Fix $s, d \in \mathbb{N}$. There exists a distribution over inputs $\mathbf{A}, \mathbf{B} \subset \{0,1\}^d$ of size s such that with probability at least 0.9 over the draw of $\mathbf{A}, \mathbf{B}$, and an execution of* ComputeEMD$(\mathbf{A}, \mathbf{B})$ *which outputs the matching $\mathbf{M}$,*

$$\sum_{(a,b)\in\mathbf{M}} \|a-b\|_1 \geq \Omega(\log s) \cdot \mathrm{EMD}(\mathbf{A}, \mathbf{B}).$$

For $s, d \in \mathbb{N}$ and $\alpha \in (0, 1/2)$, we let $\mathcal{D}_{s,d}(\alpha)$ be the distribution over pairs of subsets $(\mathbf{A}, \mathbf{B})$ with $\mathbf{A}, \mathbf{B} \subset \{0,1\}^d$ of $|\mathbf{A}| = |\mathbf{B}| = s$ given by the following procedure (think of α as being set to $1/\log s$):

- For $t = 1, \ldots, s$, we generate the pair $(\boldsymbol{a}^{(t)}, \boldsymbol{b}^{(t)})$ where $\boldsymbol{a}^{(t)}, \boldsymbol{b}^{(t)} \in \{0,1\}^d$ are sampled by letting, for each $i \in [d]$,

$$\boldsymbol{a}_i^{(t)} \sim \{0,1\} \qquad \text{and} \qquad \boldsymbol{b}_i^{(t)} \sim \begin{cases} \boldsymbol{a}_i^{(t)} & \text{w.p } 1-\alpha \\ 1-\boldsymbol{a}_i^{(t)} & \text{w.p } \alpha \end{cases}. \tag{52}$$

- We let $\mathbf{A} = \left\{\boldsymbol{a}^{(1)}, \ldots, \boldsymbol{a}^{(s)}\right\}$ and $\mathbf{B} = \left\{\boldsymbol{b}^{(1)}, \ldots, \boldsymbol{b}^{(s)}\right\}$.

Notice that for $\alpha \in (0, 1/2)$, we have

$$\mathop{\mathbf{E}}_{(\mathbf{A},\mathbf{B})\sim\mathcal{D}_{s,d}(\alpha)}[\mathrm{EMD}(\mathbf{A},\mathbf{B})] \le \sum_{t=1}^{s} \mathop{\mathbf{E}}_{(\boldsymbol{a}^{(t)},\boldsymbol{b}^{(t)})}\left[\left\|\boldsymbol{a}^{(t)} - \boldsymbol{b}^{(t)}\right\|_1\right] = sd\alpha,$$

and by Markov's inequality, $\mathrm{EMD}(\mathbf{A},\mathbf{B}) \le 100sd\alpha$ with probability at least 0.99.

On the other hand, let $\mathbf{T}$ be the (random) binary tree of depth h naturally produced by the execution of ComputeEMD$(\mathbf{A},\mathbf{B})$, and let $\mathbf{M}$ be the matching between $\mathbf{A}$ and $\mathbf{B}$ that ComputeEMD$(\mathbf{A},\mathbf{B})$ outputs. Fix $t \in [s]$, and consider the probability, over the randomness in $\mathbf{A},\mathbf{B}$ and the execution of ComputeEMD$(\mathbf{A},\mathbf{B})$ that $\boldsymbol{a}^{(t)}$ is *not* matched to $\boldsymbol{b}^{(t)}$. Notice that this occurs whenever the following event $\boldsymbol{\mathcal{E}}_t$ occurs: there exists a depth $j \in \{0,\dots,h-1\}$ such that

- At depth j, the two points $\boldsymbol{a}^{(t)}$ and $\boldsymbol{b}^{(t)}$ are split in the recursion, which occurs whenever a node $\mathbf{v}$ of $\mathbf{T}$ at depth j, corresponding to an execution of ComputeEMD$(\mathbf{A}^{(\mathbf{v})},\mathbf{B}^{(\mathbf{v})})$ with $\boldsymbol{a}^{(t)} \in \mathbf{A}^{(\mathbf{v})}$ and $\boldsymbol{b}^{(t)} \in \mathbf{B}^{(\mathbf{v})}$ samples a coordinate $\boldsymbol{i} \sim [d]$ where $\boldsymbol{a}^{(t)}_{\boldsymbol{i}} = 1$ and $\boldsymbol{b}^{(t)}_{\boldsymbol{i}} = 0$.
- Furthermore, considering the subsets which are split

$$\mathbf{A}_0^{(\mathbf{v})} = \left\{a \in \mathbf{A}^{(\mathbf{v})} : a_{\boldsymbol{i}} = 0\right\} \qquad \mathbf{A}_1^{(\mathbf{v})} = \left\{a \in \mathbf{A}^{(\mathbf{v})} : a_{\boldsymbol{i}} = 1\right\}$$
$$\mathbf{B}_0^{(\mathbf{v})} = \left\{b \in \mathbf{B}^{(\mathbf{v})} : b_{\boldsymbol{i}} = 0\right\} \qquad \mathbf{B}_1^{(\mathbf{v})} = \left\{b \in \mathbf{B}^{(\mathbf{v})} : b_{\boldsymbol{i}} = 1\right\},$$

  we happen to have $|\mathbf{B}_1^{(\mathbf{v})}| \ge |\mathbf{A}_1^{(\mathbf{v})}|$.

In order to see why this forces $\mathbf{M}(\boldsymbol{a}^{(t)}) \neq \boldsymbol{b}^{(t)}$, notice that $\boldsymbol{a}^{(t)} \in \mathbf{A}_1^{(\mathbf{v})}$ by definition that $\boldsymbol{i}$ satisfies $\boldsymbol{a}^{(t)}_{\boldsymbol{i}} = 1$, yet $\boldsymbol{b}^{(t)}_{\boldsymbol{i}} = 0$, so that $\boldsymbol{b}^{(t)} \notin \mathbf{B}_1^{(\mathbf{v})}$. Notice that since ComputeEMD$(\mathbf{A}_1^{(\mathbf{v})},\mathbf{B}_1^{(\mathbf{v})})$ returns a maximal matching $\mathbf{M}^{(\mathbf{v},1)}$ between $\mathbf{A}_1^{(\mathbf{v})}$ and $\mathbf{B}_1^{(\mathbf{v})}$. Since $|\mathbf{B}_1^{(\mathbf{v})}| \ge |\mathbf{A}_1^{(\mathbf{v})}|$, $\boldsymbol{a}^{(t)}$ participates in the matching $\mathbf{M}^{(\mathbf{v},1)}$, and hence is not matched with $\boldsymbol{b}^{(t)}$. In order to lower bound this probability, consider the following sampling process:

1. We first sample the pair of points $(\boldsymbol{a}^{(t)},\boldsymbol{b}^{(t)})$ according to (52).
2. We then sample the tree $\mathbf{T}$.
3. We sample $\ell \in [s-1]$ pairs of points $(\boldsymbol{a}^{(\ell)},\boldsymbol{b}^{(\ell)})$ similarly to (52).

Consider a fixed $(\boldsymbol{a}^{(t)},\boldsymbol{b}^{(t)})$, as well as a fixed sequence of coordinates which are sampled $\boldsymbol{i}_1,\dots,\boldsymbol{i}_j \in [d]$ such that

$$\boldsymbol{a}^{(t)}_{\boldsymbol{i}_k} = \boldsymbol{b}^{(t)}_{\boldsymbol{i}_k} \qquad \text{for all } k < j, \text{ and} \qquad \boldsymbol{a}^{(t)}_{\boldsymbol{i}_j} = 1, \boldsymbol{b}^{(t)}_{\boldsymbol{i}_j} = 0.$$

We have then

$$\mathop{\mathbf{Pr}}_{(\boldsymbol{a}^{(\ell)}\boldsymbol{b}^{(\ell)})}\left[\boldsymbol{b}^{(\ell)} \in \mathbf{B}_1^{(\mathbf{v})} \wedge \boldsymbol{a}^{(\ell)} \notin \mathbf{A}_1^{(\mathbf{v})}\right] = \frac{1}{2^j}\left(1-(1-\alpha)^j\right),$$

and since $\boldsymbol{b}^{(\ell)}$ and $\boldsymbol{a}^{(\ell)}$ are symmetric,

$$\mathop{\mathbf{Pr}}_{(\boldsymbol{a}^{(\ell)},\boldsymbol{b}^{(\ell)})}\left[\boldsymbol{a}^{(\ell)} \in \mathbf{A}_1^{(\mathbf{v})} \wedge \boldsymbol{b}^{(\ell)} \notin \mathbf{B}_1^{(\mathbf{v})}\right] = \frac{1}{2^j}\left(1-(1-\alpha)^j\right).$$

We note that event $\boldsymbol{\mathcal{E}}_t$ occurs if $(\boldsymbol{a}^{(t)}, \boldsymbol{b}^{(t)})$ are split at depth j at a coordinate $\boldsymbol{i}$ with $\boldsymbol{a}_{\boldsymbol{i}}^{(t)} = 1$, and when there exists a unique pair $(\boldsymbol{a}^{(\ell)}, \boldsymbol{b}^{(\ell)})$ which satisfy $\boldsymbol{b}^{(\ell)} \in \mathbf{B}_1^{(\mathbf{v})}$ and $\boldsymbol{a}^{(\ell)} \notin \mathbf{A}_1^{(\mathbf{v})}$. Hence,

$$\begin{aligned}&\mathbf{Pr}\left[\boldsymbol{\mathcal{E}}_t\right] \qquad (53)\\ &\geq \mathop{\mathbf{E}}_{(\boldsymbol{a}^{(t)}, \boldsymbol{b}^{(t)})}\left[\left(1 - \frac{\|\boldsymbol{a}^{(t)} - \boldsymbol{b}^{(t)}\|_1}{d}\right)^{j-1} \frac{\|\boldsymbol{a}^{(t)} - \boldsymbol{b}^{(t)}\|_1}{d} \left(\frac{s-1}{2^j}(1-(1-\alpha)^j)\right)\left(1 - \frac{1}{2^{j-1}}\left(1-(1-\alpha)^j\right)\right)^{s-2}\right].\end{aligned}$$

If we consider $j = \lfloor \log_2 s \rfloor$ and $\alpha = \frac{1}{\log_2 s}$, we have

$$\frac{s-1}{2^j}\left(1-(1-\alpha)^j\right)\left(1 - \frac{1}{2^{j-1}}\left(1-(1-\alpha)^j\right)\right)^{s-2} = \Omega(1),$$

so the probability in (53) is at least

$$\Omega(1) \cdot \mathop{\mathbf{E}}_{(\boldsymbol{a}^{(t)}, \boldsymbol{b}^{(t)})}\left[\left(1 - \frac{\|\boldsymbol{a}^{(t)} - \boldsymbol{b}^{(t)}\|_1}{d}\right)^{j-1} \frac{\|\boldsymbol{a}^{(t)} - \boldsymbol{b}^{(t)}\|_1}{d}\right] = \Omega(1),$$

since $\|\boldsymbol{a}^{(t)} - \boldsymbol{b}^{(t)}\|_1$ is distributed as $\mathrm{Bin}(d, \alpha)$, and $\alpha = \frac{1}{\log_2 s}$, and $j = \lfloor \log_2 s \rfloor$. If we let

$$\boldsymbol{m} = \min_{t_1 \neq t_2} \|\boldsymbol{a}^{(t_1)} - \boldsymbol{b}^{(t_2)}\|_1,$$

then, we have that $\mathbf{M}$ is a matching of cost at least $\boldsymbol{m}$ times the number of pairs $t \in [s]$ where $\boldsymbol{\mathcal{E}}_t$ occurs, and each $\boldsymbol{\mathcal{E}}_t$ occurs with constant probability. By Markov's inequality, with probability 0.99 over the draw of $\mathbf{A}, \mathbf{B}$ and ComputeEMD$(\mathbf{A}, \mathbf{B})$, there are $\Omega(s)$ indices $t \in [s]$ where $\boldsymbol{\mathcal{E}}_t$ occurs. Furthermore, since for any $t_1 \neq t_2$, $\boldsymbol{a}^{(t_1)}$ and $\boldsymbol{b}^{(t_2)}$ are distributed as uniformly random points, we have that with a Chernoff bound $\boldsymbol{m} \geq \Omega(d)$ with probability at least 0.99 whenever $d \geq c_0 \log s$ (for a large fixed constant c_0).

Putting everything together with a union bound, we have that with probability at least 0.97 over the draw of $\mathbf{A}, \mathbf{B}$ and ComputeEMD$(\mathbf{A}, \mathbf{B})$, $\mathrm{EMD}(\mathbf{A}, \mathbf{B}) \leq 100sd/\log_2 s$, yet the matching output has cost at least $\Omega(sd)$.

C Lower Bound for Quadtree Approximation via Tree Embeddings

In this section, we show that the analysis in Section A is tight. This, along with Theorem 1, demonstrates that the approximation given by evaluating the matching produced by Quadtree is strictly better in the original metric, rather than the tree metric, demonstrating the necessity of considering the former. Specifically, let $\mathbf{T}_0$ be the execution tree described the beginning of Section 3, where we execute the algorithm for at most $2d$ rounds of recursion. We assign weights to the edges, where an edge connecting a node at depth i and $i+1$ is given weight $d/(i+1)^2$. This defines a tree metric $(A \cup B, d_{\mathbf{T}_0})$ given by the sum of weights over the paths connecting two points in the tree. We remark that the following analysis applies in a straightforward way to the

depth $O(\log d)$ *compressed* quadtree also described in Section A, which is the same tree considered in [AIK08].

Let $\mathbf{M}_T(A, B) \subset A \times B$ be any the greedy bottom-up matching. By Claim A.1, we know that $\mathbf{M}_T$ is an optimal cost matching in the tree metric on T_0. Fix s, d larger than some constant, so that $d = s^{\Theta(1)}$ are polynomial related, and let $d^{.01} < \alpha < d^{.99}$ be a parameter which decides the cost of $\mathrm{EMD}(A, B)$. We will define two distributions, $\mathcal{D}_1$, and $\mathcal{D}_2$, over pairs of multisets $A, B \subset \{0,1\}^d$ of size s, such that

$$\Pr_{(A,B)\sim\mathcal{D}_1,\mathbf{T}_0}[\mathbf{Cost}(\mathbf{M}_{\mathbf{T}_0}(A,B)) < \frac{c}{\log s} \cdot \mathrm{EMD}(A,B)] > 99/100$$

and

$$\Pr_{(A,B)\sim\mathcal{D}_2,\mathbf{T}_0}[\mathbf{Cost}(\mathbf{M}_{\mathbf{T}_0}(A,B)) > c \log s \cdot \mathrm{EMD}(A,B)] > 99/100$$

and finally

$$\Pr_{(A,B)\sim\mathcal{D}_1}[\mathrm{EMD}(A,B) = (1 \pm 1/3)\alpha s] > 1 - 1/s$$
$$\Pr_{(A,B)\sim\mathcal{D}_2}[\mathrm{EMD}(A,B) = \alpha s] > 1 - 1/s$$

for some fixed constants c, C'. This will demonstrate that the cost of the matching given by the embedding into the tree metric is a $\Omega(\log^2 s)$ approximation.

The First Distribution. We first describe $\mathcal{D}_1$. To draw $(A, B) \sim \mathcal{D}_1$, we set $d' = \alpha$, and pick $2s$ uniformly random points $a'_1, a'_2, \ldots, a'_s, b'_1, b'_2, \ldots, b'_s \sim \{0,1\}^{d'}$, and set $A = \{a_1, \ldots, a_s\}, B = \{b_1, \ldots, b_s\}$, where a_i is a'_i padded with 0's, and similarly for b_i.

The Second Distribution. We now describe the second distribution $\mathcal{D}_2$. To draw $(A, B) \sim \mathcal{D}_2$, we set $a_1, \ldots, a_s \sim \{0,1\}^d$ uniformly at random. Then, for each $i \in [s]$, we set b_i to be a uniformly random point at distance α from a_i. In other words, given a_i, the point b_i is obtained by selecting a random subset of α coordinates in $[d]$, and having b_i disagree with a_i on these coordinates (and agree elsewhere).

Proposition C.1. *There exists a fixed constant $c > 0$ such that*

$$\Pr_{(A,B)\sim\mathcal{D}_1,\mathbf{T}_0}\left[\mathbf{Cost}(\mathbf{M}_{\mathbf{T}_0}(A,B)) < \frac{c}{\log s}\,\mathrm{EMD}(A,B)\right] > 99/100$$

And moreover,

$$\frac{\alpha s}{3} < \mathrm{EMD}(A,B) \leq \frac{2\alpha s}{3}$$

with probability at least $1 - 1/s$ over the draw of $(A, B) \sim \mathcal{D}_1$.

Proof: Notice that for all a, b, we have $d'/3 \leq d_{\mathbf{T}_0}(a, b) \leq 2d'/3$ with probability $1 - 2^{-\Omega(d)}$, thus by a union bound this holds for all $s^2 = \mathrm{poly}(d)$ pairs with probability at least $1 - 1/s^2$. In this case, $\alpha s/3 < \mathrm{EMD}(A, B) \leq (2/3)\alpha s$ with probability $1 - 1/s$.

Now consider any fixed point $a \in A$. We compute the expected cost of matching a in $\mathbf{M}_{\mathbf{T}_0}(A, B)$. Specifically, let $\mathbf{Cost}(a) = d_{\mathbf{T}_0}(a, b_a)$, where $(a, b_a) \in \mathbf{M}_{\mathbf{T}_0}$. To do this, we must first define how the matching $\mathbf{M}_{\mathbf{T}_0}(A, B)$ breaks ties. To do this, in the algorithm of Figure 1, we randomly choose the maximal partial matching between the unmatched points in a recursive call to ComputeEMD.

Now let $X = A \cup B$, and begin generating the randomness needed for the branch which handles the entire set of points in X. Namely, consider following down the entire set of points X down $\mathbf{T}_0$, until the first vertex r which splits X up into two non-empty subsets in its children. Namely, $A_r \cup B_r = X$, but at r an index $\boldsymbol{i} \in [d']$ is sampled which splits X. First note that we expect the depth of r to be $\ell = d/\alpha$, since $d'/d = 1/\sqrt{d}$. Moreover, with probability $199/200$ we have $d/(2000\alpha) < \ell < 2000d/\alpha$. Call this event $\mathcal{E}_0$, and condition on this now, which does not fix the affect randomness used in any of the subtree of r. Conditioned on $\mathcal{E}_0$, we have that $\mathbf{Cost}(a)$ conditioned on $\mathcal{E}_0$ is at most $\sum_{i=\ell}^{2d} \frac{d}{i^2} = \Theta(d/\ell) = \Theta(\alpha)$, since all points are matched below depth ℓ.

Now consider the level $\ell_1 = \ell + (1/2)\log s(d/\alpha)$. Notice the the vertex v which contains the point a at depth ℓ_1 as a descendant of r in expectation samples between $(1/2)(1 - \frac{1}{1000})\log s$ and $(1/2)(1 + \frac{1}{1000})\log s$ **unique** coordinates $\boldsymbol{i} \in [\alpha]$ on the $(1/2)\log s(d/\alpha)$ steps between r and v (we have not fixed the randomness used to draw the coordinate a yet). By independence across samples, by Chernoff bounds we have that v sampled between $(1/3)\log s$ and $(2/3)\log s$ **unique** coordinates $\boldsymbol{i} \in [\alpha]$ with probability $1 - s^{-c}$ for some fixed constant $c = \Omega(1)$. Let $\mathcal{E}_a$ be the event that this occurs. Say that it samples exactly γ unique coordinates. Now $\mathbf{E}[|A_v \cup B_v|] = (2s)/2^\gamma \geq s^{1/3}$, where the randomness is taken over the choice of random points in A, B. By Chernoff bounds applied to both the size of $|A_v|, |B_v|$, we have

$$\mathbf{Pr}[||A_v| - |B_v|| > c_1 \log(s)\sqrt{s/2^\gamma} \geq] < 1/s^c$$

for a sufficiently large constant c_1. Call the event that the above does not occur $\mathcal{E}_1$, and condition on it now. Note that conditioned on $\mathcal{E}_1$, only a $c_1 \log s\sqrt{2^\gamma/s} < s^{-1/7}$ fraction of the points in v remain unmatched. Since the distribution of the path a point $x \in A_v \cup B_v$ takes in the subtree of v is identical to every other $x' \in A_v \cup B_v$ even conditioned on $\mathcal{E}_0, \mathcal{E}_1, \mathcal{E}_a$, it follows that

$$\begin{aligned} \mathbf{E}[\mathbf{Cost}(a) \mid \mathcal{E}_0, \mathcal{E}_1, \mathcal{E}_a] &\leq O(\frac{d}{\ell_1}) + s^{-1/7}O(\frac{d}{\ell}) \\ &\leq O(\frac{\alpha}{\log s}) \end{aligned} \tag{54}$$

Thus

$$\begin{aligned} \mathbf{E}[\mathbf{Cost}(a) \mid \mathcal{E}_0] &\leq O(\frac{\alpha}{\log s}) + s^{-c}O(\alpha) \\ &\leq O(\frac{\alpha}{\log s}) \end{aligned} \tag{55}$$

Then by Markov's inequality, using that $\alpha s/3 < \mathrm{EMD}(A, B) \leq (2/3)\alpha s$ with probability $1 - 1/s$, we have

$$\mathbf{Pr}[\sum_{a\in A}\mathbf{Cost}(a) > \frac{c}{\log s}\,\mathrm{EMD}(A,B) \mid \mathcal{E}_0] \leq 10^{-4}$$

By a union bound:

$$\begin{aligned}\mathop{\mathbf{Pr}}_{(A,B)\sim\mathcal{D}_1,\mathbf{T}_0}\left[\mathbf{Cost}(\mathbf{M}_{\mathbf{T}_0}(A,B)) < \frac{c}{\log s}\,\mathrm{EMD}(A,B)\right] &> 1 - 10^{-4} - 1/200 - O(1/s)\\ &> 99/100\end{aligned} \tag{56}$$

which completes the proof.

■

Proposition C.2. *There exists a fixed constant $c > 0$ such that*

$$\mathop{\mathbf{Pr}}_{(A,B)\sim\mathcal{D}_2,\mathbf{T}_0}[\mathbf{Cost}(\mathbf{M}_{\mathbf{T}_0}(A,B)) < c\log s\cdot\mathrm{EMD}(A,B)] > 99/100$$

And moreover,

$$\mathrm{EMD}(A,B) = \alpha\cdot s$$

with probability at least $1-1/s$ over the draw of $(A,B)\sim\mathcal{D}_2$.

Proof: Set $\beta = 20\log s$. We first claim that the event $\mathcal{E}_0$, which states that every non-empty vertex $v\in\mathbf{T}_0$ at depth β satisfies either $A_v\cup B_v = \{a_i,b_i\}$, $A_v\cup B_v = \{a_i\}$, or $A_v\cup B_v = \{b_i\}$ for some $i\in[s]$, occurs with probability $1-2/s^2$. To see this, note that first for each $i\neq j$, $d(a_i,b_j), d(a_i,a_j), d(b_i,b_j)$ are each at least $d/3$ with probability $1-2^{-\Omega(d)}$, and we can the union bound over all s^2 pairs so that this holds for all $i\neq j$ with probability $1-1/s^2$. Given this, for any a_i to collide at depth β with either a_j or b_j when $i\neq j$, we would need to avoid the set of $d/3$ points where they do not agree. The probability that this occurs is $(2/3)^\beta < 1/s^4$, and we can union bound over all s^2 pairs again for this to hold with probability $1-1/s^2$. We condition on $\mathcal{E}_0$ now. First note that the probability that a_i is split from b_i at or before level β is at most $\frac{3\beta}{d}$. Thus the expected number of $i\in[s]$ for which this occurs is $\frac{3s\beta}{d}$, and is at most $\frac{30^4 s\beta}{d}$ with probability $1-10^{-4}$. Call this latter event $\mathcal{E}_1$, and let $S\subset[s]$ be the set of points i for which a_i,b_i are together, with no other points, in their node at depth β. Conditioned on $\mathcal{E}_0$ and $\mathcal{E}_1$, we have $|S| > s - \frac{30^4 s\beta}{d}$

Now for $j = \log(\beta),\ldots,\log(d/\alpha)$, let $S_j\subset S$ be the set of $i\in[s]$ for which a_i and b_i split before level 2^j. We have that $\mathbf{E}[|S_j|] > \frac{s2^j\alpha}{10d}$. Notice that since each branch of the quadtree is independent, and all (a_i,b_i) are together in their own node at depth β when $i\in S$, we can apply Chernoff bounds to obtain $|S_j| > \frac{s2^j\alpha}{20d}$ with probability at least $1-1/s^2$ for every $j > \log(d/\alpha) - (1/10)\log s$, and we can then union bound over the set of such j. Note that for each $i\in S_j$, the point a_i pays a cost of at least $\Theta(\frac{d}{2^j})$, thus the total cost of all points in S_j is at least $\Omega(s\alpha)$, and summing over all $\log(d/\alpha) - (1/10)\log s < j < \log(d/\alpha)$, we obtain $\mathbf{Cost}(\mathbf{M}_{\mathbf{T}_0}(A,B)) = \Omega(\log s\alpha)$.

We finally show that $\mathrm{EMD}(A,B) = \alpha\cdot s$ with probability at least $1-1/(2s)$. To see this, note that conditioned on $\mathcal{E}_0$, the optimal matching is (a_i,b_i). Since $d(a_i,b_i) = \alpha$, this completes the proof. ■

D Sampling with Meta-data

In this section, We demonstrate how the tools developed in Section 6 can be easily applied to obtain a linear sketching algorithm for the problem of *Sampling with Meta-Data*, which we now define.

Definition D.1. *Fix any constant $c > 1$, and fix k, n with $k \leq n^c$, and let $\epsilon, \delta > 0$. Fix any $x \in \mathbb{R}^n$, and meta-data vectors $\lambda_1, \ldots, \lambda_n \in \mathbb{R}^k$, such that the coordinates of x and all λ_i are can be stored in $O(\log n)$ bits. In the sampling with meta-data problem, we must sample $i^* \sim [n]$ from the distribution*

$$\mathbf{Pr}[i^* = i] = (1 \pm \epsilon)|x_i|/\|x\|_1 \pm n^{-c}$$

for all $i \in [n]$. Conditioned on returning $i^* = i \in [n]$, the algorithm must then return an approximation $\widehat{\lambda}_i$ to λ_i with probability $1 - \delta$.

Remark 15. *A natural generalization of Definition D.1 is to ℓ_p sampler for $p \in (0, 2]$, where we want to sample $i \in [n]$ proportional to $|x_i|^p$. We remark that the below precision sampling framework easily generalizes to solving ℓ_p sampling with meta-data, by replacing the scalings $1/t_i$ by $1/t_i^{1/p}$, and appropriately modifying Lemma 6.5 to have error $\epsilon\|x\|_p$ instead of $\epsilon\|x\|_1$. One can then apply the ℓ_p variant of Lemma 6.6 from Proposition 1 of [JW18], and the remainder of the proof follows similarly as below.*

We given a linear sketching algorithm for the problem in Definition D.1. The algorithm is similar, and simpler, to the algorithm from Section 6. Specifically, given $x \in \mathbb{R}^n, \lambda_1, \ldots, \lambda_n \in \mathbb{R}^k$, we construct a matrix $\mathbf{X} \in \mathbb{R}^{n \times (k+1)}$, where for each $i \in [n]$ the row $\mathbf{X}_{i,*} = [x_i, \lambda_{i,1}, \lambda_{i,2}, \ldots, \lambda_{i,k}]$. We then draw a random matrix $\mathbf{D} \in \mathbb{R}^{n \times n}$, where $\mathbf{D}_{i,i} = 1/t_i$ and $\{t_i\}_{i \in [n]}$ are i.i.d. exponential random variables. We then draw a random count-sketch matrix (Theorem 11) $\mathbf{S}$ and compute the sketch $\mathbf{SDX}$, and recover an estimate $\tilde{\mathbf{Z}}$ of $\mathbf{DX}$ from Count-sketch. To sample a coordinate, we output $i^* = \arg\max_i \tilde{\mathbf{Z}}_{i,1}$, and set our estimate $(\widehat{\lambda}_{i^*})_j = t_{i^*} \tilde{\mathbf{Z}}_{i,j+1}$. This simple algorithm obtains the following guarantee.

Theorem 16. *Fix any $\epsilon > 0$ and constant $c > 1$. Then given $x \in \mathbb{R}^n$ and $\lambda_1, \ldots, \lambda_n \in \mathbb{R}^k$, as in Definition D.1, the above linear sketching algorithm samples $i \in [n]$ with probability $(1 \pm \epsilon)\frac{|x_i|}{\|x\|_1} \pm n^{-c}$. Once i is sampled, it returns a value $\widehat{\lambda}_i$ such that $\|\widehat{\lambda}_i - \lambda_i\|_1 \leq \epsilon |x_i| \frac{\sum_{j \in [n]} \|\lambda_j\|_1}{\|x\|_1}$ with probability $1 - n^{-c}$. The total space required to store the sketch is $O(\frac{k}{\epsilon} \log^3(n))$ bits.*

Proof: Let $\mathbf{Z} = \mathbf{DX}$. We instantiate count-sketch with parameter $\eta = \Theta(\sqrt{\epsilon / \log(n)})$ with a small enough constant. By Theorem 11, for every $j \in [n]$ we have $|\tilde{\mathbf{Z}}_{i,1} - \mathbf{Z}_{i,1}| < \eta \|(\mathbf{Z}_{*,1})_{-1/\eta^2}\|_2$ with probability $1 - n^{-2c}$. By Lemma 6.6, we have $\|(\mathbf{Z}_{*,1})_{-1/\eta^2}\|_2 \leq \eta \|\mathbf{X}_{*,1}\|_1$ with probability $1 - n^{-2c}$, and thus $|\tilde{\mathbf{Z}}_{i,1} - \mathbf{Z}_{i,1}| < \eta^2 \|\mathbf{X}_{*,1}\|_1$, so by Lemma 6.5, the coordinate i^* satisfies

$$\mathbf{Pr}[i^* = i] = (1 \pm \epsilon)\frac{|x_i|}{\|x\|_1} \pm n^{-c}$$

for all $i \in [n]$, where we used the fact that the n^{-2c} failure probabilities of the above events can be absorbed into the additive n^{-c} error of the sampler. Now recall by Fact 6.4 that $\max_i \mathbf{Z}_{i,1} = \|\mathbf{X}_{*,1}\|_1 / E$, where E is an independent exponential random variable. With probability $1 - n^{-c}$,

using the tails of exponential variables we have $\max_i \mathbf{Z}_{i,1} > \|\mathbf{X}_{*,1}\|_1/(10c\log s)$, which we condition on now. Notice that given this, since our error satisfies $|\tilde{\mathbf{Z}}_{i,1} - \mathbf{Z}_{i,1}| < \eta^2\|\mathbf{X}_{*,1}\|_1$, we have $\mathbf{Z}_{i^*,1} > \|\mathbf{X}_{*,1}\|_1/(20c\log s)$, from which it follows by definition that $t_i < \frac{20c\log s|x_i|}{\|x\|_1}$.

Now consider any coordinate $j \in [k]$. By the count-sketch error, we have $|\tilde{\mathbf{Z}}_{i,j+1} - \mathbf{Z}_{i,j+1}| < \eta\|(\mathbf{Z}_{*,j+1})_{-1/\eta^2}\|_2$, and moreover by Lemma 6.6, we have $\|(\mathbf{Z}_{*,1})_{-1/\eta^2}\|_2 \le \eta\|\mathbf{X}_{*,j+1}\|_1 = \eta\sum_{i\in[n]}|\lambda_{i,j}|$ with probability $1 - n^{-2c}$. Thus

$$\begin{aligned} |t_i\tilde{\mathbf{Z}}_{i,j+1} - t_i\mathbf{Z}_{i,j+1}| &< t_i\eta^2\sum_{i\in[n]}|\lambda_{i,j}| \\ &\le \eta^2\frac{20c\log s|x_i|\sum_{i\in[n]}|\lambda_{i,j}|}{\|x\|_1} \\ &\le \epsilon\frac{|x_i|\sum_{i\in[n]}|\lambda_{i,j}|}{\|x\|_1} \end{aligned} \tag{57}$$

Summing over all $j \in [k]$ (after a union bound over each $j \in [k]$) completes the proof of the error. For the space, note that $\mathbf{SDX}$ had $O(1/\eta^2\log n) = O(\log^2 n/\epsilon)$ rows, each of which has $k+1$ columns. Note that we can truncating the exponential to be $O(\log n)$ bit discrete values, which introduces an additive n^{-10c} to each coordinate of $\mathbf{SDX}$ (using that the values of $\mathbf{X}$ are bounded by $n^c = \text{poly}(n)$), which can be absorbed into the additive error from count-sketch without increasing the total error by more than a constant. Moreover, assuming that the coordinates of x and all λ_i are integers bounded by $\text{poly}(n)$ in magnitude, since the entries of $\mathbf{S}$ are contained in $\{0, 1, -1\}$, each entry of $\mathbf{SDX}$ can be stored in $O(\log n)$ bits of space, which completes the proof. ■

E Embedding ℓ_p^d into $\{0,1\}^{d'}$

Lemma E.1. *Let $p \in (1, 2]$. There exists a distribution $\mathcal{D}_p$ supported on $\mathbb{R}$ which exhibits p-stability: for any $d \in \mathbb{N}$ and any vector $x \in \mathbb{R}^d$, the random variables $\mathbf{Y}_1$ and $\mathbf{Y}_2$ given by*

$$\begin{aligned} \mathbf{Y}_1 &= \sum_{i=1}^d x_i\boldsymbol{z}_i \qquad \textit{for } \boldsymbol{z}_1,\dots,\boldsymbol{z}_d \sim \mathcal{D}_p \textit{ independently,} \\ \mathbf{Y}_2 &= \boldsymbol{z}\cdot\|x\|_p \qquad \textit{for } \boldsymbol{z} \sim \mathcal{D}_p, \end{aligned}$$

are equal in distribution, and $\mathbf{E}_{\boldsymbol{z}\sim\mathcal{D}_p}[|\boldsymbol{z}|]$ is at most a constant C_p.

Consider fixed $R \in \mathbb{R}^{\ge 0}$ which is a power of 2, and for $p \in [1, 2]$, let $\boldsymbol{f}\colon \mathbb{R}^d \to \{0,1\}$ be the randomized function given by sampling the following random variables

- $\boldsymbol{z} = (\boldsymbol{z}_1,\dots,\boldsymbol{z}_d)$ where $\boldsymbol{z}_1,\dots,\boldsymbol{z}_d \sim \mathcal{D}_p$. In the case $p = 1$, we let $\mathcal{D}_1$ be the distribution where $\boldsymbol{z}_i \sim \mathcal{D}_1$ is set to 1 with probability $1/d$ and 0 otherwise.
- $\boldsymbol{h} \sim [0, R]$, and

- a uniformly random function $\boldsymbol{g}\colon \mathbb{Z} \to \{0,1\}$.

and we evaluate

$$\boldsymbol{f}(x) = \boldsymbol{g}\left(\left\lfloor \frac{\sum_{i=1}^d x_i \boldsymbol{z}_i - \boldsymbol{h}}{R} \right\rfloor\right) \in \{0,1\}.$$

If $x, y \in \mathbb{R}^d$ with $\|x-y\|_p \le R/t_p$ (for a parameter t_p which we specify later). When $p=1$, we consider $\|x-y\|_\infty \le R/t_1$. Then

$$\Pr_{\boldsymbol{z},\boldsymbol{h},\boldsymbol{g}}\left[\boldsymbol{f}(x) \neq \boldsymbol{f}(y)\right] = \frac{1}{2} \cdot \Pr_{\boldsymbol{z},\boldsymbol{h}}\left[\left\lfloor \frac{\sum_{i=1}^d x_i \boldsymbol{z}_i - \boldsymbol{h}}{R} \right\rfloor \neq \left\lfloor \frac{\sum_{i=1}^d y_i \boldsymbol{z}_i - \boldsymbol{h}}{R} \right\rfloor\right].$$

Consider a fixed $\boldsymbol{z} = (\boldsymbol{z}_1, \ldots, \boldsymbol{z}_d)$ and let

$$\boldsymbol{w} = \sum_{i=1}^d (x_i - y_i)\boldsymbol{z}_i.$$

Then, we have

$$\Pr_{\boldsymbol{h}\sim[0,R]}\left[\left\lfloor \frac{\sum_{i=1}^d x_i \boldsymbol{z}_i - \boldsymbol{h}}{R} \right\rfloor \neq \left\lfloor \frac{\sum_{i=1}^d y_i \boldsymbol{z}_i - \boldsymbol{h}}{R} \right\rfloor\right] = \max\left\{\frac{|\boldsymbol{w}|}{R}, 1\right\},$$

so that

$$\Pr_{\boldsymbol{z},\boldsymbol{h},\boldsymbol{g}}\left[\boldsymbol{f}(x) \neq \boldsymbol{f}(y)\right] = \frac{1}{2}\mathop{\mathbf{E}}_{\boldsymbol{z}}\left[\max\left\{\frac{|\boldsymbol{w}|}{R}, 1\right\}\right].$$

When $p \in (1,2]$, we notice that by Jensen's inequality and Lemma E.1,

$$\mathop{\mathbf{E}}_{\boldsymbol{z}}\left[\max\left\{\frac{|\boldsymbol{w}|}{R}, 1\right\}\right] \le \max\left\{\frac{C_p\|x-y\|_p}{R}, 1\right\} = \frac{C_p\|x-y\|_p}{R},$$

and similarly, $\mathbf{E}_{\boldsymbol{z}}\left[\max\{|\boldsymbol{w}|/R, 1\}\right] \le \|x-y\|_1/(dR)$ when $p=1$. Furthermore, for $p \in (1,2]$, we lower bound the above quantity by

$$\mathop{\mathbf{E}}_{\boldsymbol{z}}\left[\max\left\{\frac{|\boldsymbol{w}|}{R}, 1\right\}\right] \ge \frac{\|x-y\|_p}{R} \cdot \mathop{\mathbf{E}}_{\boldsymbol{z}\sim\mathcal{D}_p}\left[\max\left\{|\boldsymbol{z}|, t_p\right\}\right] \ge \frac{\|x-y\|_p}{R} \cdot \frac{C_p}{2},$$

since $\mathbf{E}_{\boldsymbol{z}\sim\mathcal{D}_p}\left[\max\{|\boldsymbol{z}|, t_p\}\right] \to C_p$ as $t_p \to \infty$, which means that for some constant setting of high enough t_p, we obtain the above inequality. In particular, for $p \in (1,2]$, we may choose t_p to be a large enough constant depending only on p. When $p=1$,

$$\mathop{\mathbf{E}}_{\boldsymbol{z}}\left[\max\left\{\frac{|\boldsymbol{w}|}{R}, 1\right\}\right] \ge \sum_{i=1}^d \frac{1}{d}\left(1-\frac{1}{d}\right)^{d-1} \max\left\{\frac{|x_i-y_i|}{R}, 1\right\} = \frac{1}{d}\left(1-\frac{1}{d}\right)^{d-1}\frac{\|x-y\|_1}{R},$$

by setting $t_1 = 1$.

For $p \in (1,2]$, we consider concatenating m independent executions of the above construction and consider that as an embedding of $\boldsymbol{\phi}\colon \mathbb{R}^d \to \{0,1\}^m$, we have that for any $x, y \in \mathbb{R}^d$ with $\|x-y\|_p \le R/t_p$,

$$\begin{aligned}\mathop{\mathbf{E}}_{\boldsymbol{\phi}}\left[\|\boldsymbol{\phi}(x)-\boldsymbol{\phi}(y)\|_1\right] &= m \cdot \mathop{\mathbf{Pr}}_{\boldsymbol{f}}\left[\boldsymbol{f}(x) \neq \boldsymbol{f}(y)\right] \\ &\asymp m \cdot \frac{\|x-y\|_p}{R},\end{aligned}$$

where the notation $\asymp$ suppresses constant factors. Suppose $A \cup B \subset \mathbb{R}^d$ is a subset of at most $2s$ points with aspect ratio

$$\Phi \stackrel{\text{def}}{=} \frac{\max_{x,y\in A\cup B}\|x-y\|_p}{\min_{\substack{x,y\in A\cup B\\ x\neq y}}\|x-y\|_p}.$$

Then, we let $R = t_p \cdot \max_{x,y\in A\cup B}\|x-y\|_p$, and

$$m = O\left(t_p \Phi \log s\right).$$

For any fixed setting of $x, y \in A \cup B$, the distance between $\boldsymbol{\phi}(x)$ and $\boldsymbol{\phi}(y)$ is the number of independent of executions of $\boldsymbol{f}$ where $\boldsymbol{f}(x) \neq \boldsymbol{f}(y)$. Since executions are independent, we apply a Chernoff bound to say that with probability $1 - 1/s^{100}$ every $x, y \in A \cup B$ has $\|\boldsymbol{\phi}(x) - \boldsymbol{\phi}(y)\|_1$ concentrating up to a constant factor around its expectation, and therefore, we apply a union bound over $4s^2$ many pairs of points in $A \cup B$ to conclude $\boldsymbol{\phi}$ is a constant distortion embedding. The case $p = 1$ follows similarly, except that

$$\mathop{\mathbf{E}}_{\boldsymbol{\phi}}\left[\|\boldsymbol{\phi}(x)-\boldsymbol{\phi}(y)\|_1\right] \asymp m \cdot \frac{\|x-y\|_1}{dR},$$

and $R = 2\max_{x,y\in A\cup B}\|x-y\|_\infty$. In summary, we have the following lemmas.

Lemma E.2. *Fix any $p \in (1,2]$, $n, d \in \mathbb{N}$, and a parameter $\Phi \in \mathbb{R}^{\ge 0}$. There exists a distribution $\mathcal{E}_p$ over embeddings $\boldsymbol{\phi}\colon \mathbb{R}^d \to \{0,1\}^{d'}$, where*

$$d' = O(\Phi \log n),$$

such that for any fixed set $X \subset \mathbb{R}^d$ of size at most n and aspect ratio in ℓ_p at most Φ, a draw $\boldsymbol{\phi} \sim \mathcal{E}_p$ is a constant distortion embedding of X into the hypercube with Hamming distance with probability at least $1 - 1/n^{10}$.

Lemma E.3. *Fix any $n, d \in \mathbb{N}$, and a parameter $r, R \in \mathbb{R}^{\ge 0}$. There exists a distribution $\mathcal{E}_1$ over embeddings $\boldsymbol{\phi}\colon \mathbb{R}^d \to \{0,1\}^{d'}$, where*

$$d' = O\left(\frac{dR\log n}{r}\right),$$

such that for any fixed set $X \subset \mathbb{R}^d$ of size at most n and ℓ_∞ distance at most R and ℓ_1 distance at least r, a draw $\boldsymbol{\phi} \sim \mathcal{E}_1$ is a constant distortion embedding of X into the hypercube with Hamming distance with probability at least $1 - 1/n^{10}$.

F Linear Sketching, Communication, and Streaming

In this section, we define and discuss the two-party communication, streaming, and linear sketching models.

Two-Party Communication. In the two-party communication problem, there are two parties, Alice and Bob. Alice is given as input a multi-set $A \subset \{0,1\}^d$, and Bob is also given a multi-set $B \subset \{0,1\}^d$, where $|A| = |B| = s$. Their goal is to jointly approximate the value of $\text{EMD}(A, B)$. To do this, Alice and Bob exchange messages in *rounds*, where in each round one player sends exactly one message to the other player. Without loss of generality, we assume that Alice sends a message first. Thus, in a one-round protocol, Alice sends exactly one message M_1 to Bob, and then given M_1 and his input B, Bob must output an approximation $\tilde{R}$ to $\text{EMD}(A, B)$. In a two-round protocol, Alice sends exactly one message M_1 to Bob. Given M_1 and B, Bob decides on a message M_2, and sends M_2 to Alice. After receiving M_2, Alice must then output an approximation $\tilde{R}$ to $\text{EMD}(A, B)$. We work in the *public-coin* model of communication, where Alice and Bob have access to a shared infinite random string.

A protocol $\mathcal{P}$ for the two-party Earth-Mover Distance approximation problem is the procedure by which Alice and Bob compute the messages and their output. A protocol $\mathcal{P}$ is said to be correct if it achieves a desired approximation with probability at least $2/3$ over the coin flips of $\mathcal{P}$ and the shared random string. For a protocol $\mathcal{P}$, the *communication complexity* of $\mathcal{P}$ is the maximum total length of all exchanged messages.

The Streaming Model. In the steaming model, points arrive from $A \cup B$ in a stream. Specifically, at each time step in the stream, a point $p \in \{0,1\}^d$ is inserted into the stream, along with an identifier of whether $p \in A$ or $p \in B$. At the end of the stream, the algorithm must output an approximation to $\text{EMD}(A, B)$. In the *turnstile* model of streaming, points p can also be *deleted* from the stream (not necessarily having been inserted before), so long as at the end of the stream the sets A, B defined by the stream satisfy $|A| = |B| = s$. This geometric stream can in fact be modeled as a typical data stream, where the updates are coordinate-wise updates to a "frequency vector" f. Here, we let $f \in \mathbb{R}^n$, where $n = 2 \cdot 2^d$ be a vector initalized to 0. At each time step t, the vector f received a coordinate-wise update (i_t, Δ_t), where $i_t \in [n]$ and Δ_t is a integer, which causes the change $f_{i_t} \leftarrow f_{i_t} + \Delta_t$. To see the equivalence, if we want to insert a point p into A a total of z times, we can make the update (p, z), where p indexes into $[2^d]$ in the natural way. Similarly, to add a point p into B a total of z times, we make the update $(p + 2^d, z)$, and deletions are handled similarly. We make the common assumption that $|\Delta_t| = \text{poly}(s)$ for all t, and that the length of the stream is at most $\text{poly}(s)$, so that the coordinates of the vector f can be represented with $\log(s)$ bits of space at any intermediate point in the stream.

The goal of the streaming model is to maintain a small space sketch of the vector f, so at the end of the stream an algorithm can produce a good approximation to the earth-mover distance $\text{EMD}(A, B)$. When discussing the space complexity of streaming algorithms, there are two separate notions: *working space* and *intrinsic space*. We remark that generally these two notations of space are the same (up to constant factors) for streaming algorithms, however for our purposes it will be useful to distinguish them. The *working space* of a streaming algorithm $\mathcal{A}$ is the space required

to store an update (i_t, Δ_t) in the stream and process it. The *intrinsic space* is the space which the algorithm must store *between* updates. The intrinsic space coincides with the size of a message which must be passed from one party two another, if each party, for instance, holds some fraction of the data stream. Thus, streaming computation is generally focused on the intrinsic space, which we will hereafter just refer to as the *space* of the algorithm. Notice that the working space must necessarily be sufficient to read the index i_t. In the case of EMD, i_t must be represented with d bits of space, meaning that $\Omega(d)$ is a lower bound on the working memory of a streaming algorithm in this model. However, the space complexity of streaming algorithms for EMD may be smaller than this required working space.

In the streaming model, the corresponding notation for the *public coin* model is the *random oracle* model of computation. This simply establishes that the steaming algorithm is given random access to an infinitely long string of random bits, whose size does not count against the space complexity of the algorithm. As show below, linear sketching immedately implies a streaming algorithm with the same space in the random oracle model. To remove this assumption, the usage of pseudo-random generators or limited independence is generally required. In the *one-pass* streaming model, the algorithm only sees the sequence of updates a single time, whereas in the *two-pass* model the algorithm sees the sequence of updates exactly twice, one after the other.

Linear Sketching We now recall the concept of a *linear-sketch*. Linear sketching results in algorithms both for the streaming and two-party communication models. In this model, the multiset inputs $A, B \subset \{0,1\}^d$ are implicitly encoded by a vector $f_{A,B} \in \mathbb{R}^{2 \cdot 2^d}$. A linear sketch stores only the value $\mathbf{S} \cdot f_{A,B}$, where $\mathbf{S}$ is a (possibly random) matrix with $k \ll 2^d$ rows. The algorithm then outputs an estimate of $\text{EMD}(A, B)$ given only knowledge of $\mathbf{S}f_{A,B}$ and $\mathbf{S}$. The space of a linear sketch is the space required to store $\mathbf{S}f_{A,B}$, since $\mathbf{S}$ is generated with public randomness which is not charged against that algorithm. This coincides with the space of a public coin communication protocol, or the space of a streaming algorithm in the random oracle model.

Given an update (i_t, Δ_t), one can update the sketch $\mathbf{S}f_{A,B} \leftarrow \mathbf{S}f_{A,B} + \mathbf{S}_{*,i_t}\Delta_t$. This allows a linear sketch to be maintained in a stream. Moreover, since the sketch Sf is linear, the order or sequence of updates in a stream do not matter. Given a linear sketching algorithm for earth-mover distance with sketch size $\mathbf{S}f_{A,B}$ with k rows, this yields a $O(k \log s)$ communication protocol for two-party one-round communication. This follows from the fact that the matrix $\mathbf{S}$ can be generated with shared randomness. Alice can then compute the vector f_A which is induced by her set A, and Bob can similarly compute f_B, such that $f_{A,B} = f_A + f_B$. Alice then sends $\mathbf{S}f_A$ to Bob, who can compute $\mathbf{S}f_{A,B} = \mathbf{S}f_A + \mathbf{S}f_B$, and therefore solve the communication problem.

There is a corresponding notion of linear sketching for the two-round communication and two-pass streaming models, which we call a two-round sketching algorithm. A two round sketching algorithm first (with no knowledge of $f_{A,B}$) generates a matrix $\mathbf{S}_1$ from some distribution $\mathcal{D}^1$, and computes $\mathbf{S}_1 f_{A,B}$. Then, given knowledge only of $\mathbf{S}_1$ and $\mathbf{S}_1 f_{A,B}$, it generates a *second* matrix $\mathbf{S}_2$ from a distribution $\mathcal{D}^2(\mathbf{S}_1, \mathbf{S}_1 f_{A,B})$, and computes $\mathbf{S}_2 f_{A,B}$. Finally, given as input only the values $(\mathbf{S}_1, \mathbf{S}_2, \mathbf{S}_1 f_{A,B}, \mathbf{S}_2 f_{A,B})$, it outputs an approximation to $\text{EMD}(A, B)$. The space of the algorithm is the number of bits required to store $\mathbf{S}_1 f_{A,B}$ and $\mathbf{S}_2 f_{A,B}$.

It is easy to see that a two-round linear sketching algorithm results in both a two-pass streaming algorithm and a two-round communication protocol. For the former, on the first pass the algorithm

maintains $\mathbf{S}_1 f$ and $\mathbf{S}_1$ using shared randomness. At the end of the first pass, it can generate $\mathbf{S}_2$ based on $\mathbf{S}_1 f_{A,B}$ and $\mathbf{S}_1$, and compute $\mathbf{S}_2 f_{A,B}$ as needed. For a two-round communication protocol, Alice and Bob jointly generate $\mathbf{S}_1$, then Alice sends $\mathbf{S}_1 f_A$ to Bob, who computes himself $\mathbf{S}_1 f_{A,B}$, and then using shared randomness and his knowledge of $\mathbf{S}_1, \mathbf{S}_1 f_{A,B}$ can compute $\mathbf{S}_2$. Bob then sends $\mathbf{S}_1 f_{A,B}, \mathbf{S}_2 f_B$ back to Alice, who can now fully determine $\mathbf{S}_1, \mathbf{S}_2, \mathbf{S}_1 f_{A,B}, \mathbf{S}_2 f_{A,B}$ and output the approximation.

Two-Pass Streaming. We now demonstrate that this two-round linear sketch of Section 5 can be applied to obtain a two-pass streaming algorithm in the turnstile (insertion and deletion) model. Here, a stream of at most $\text{poly}(s)$ updates arrives in the stream, where each update either inserts or deletes a point from A, or inserts or delets a point from B. Noticed that the t-th update can be modelled by coordinate-wise updates to $f_{A,B}$ of the form $(i_t, \Delta_t) \in [2 \cdot 2^d] \times \{-M, -M+1, \ldots, M\}$, where $M = \text{poly}(s)$, causing the change $(f_{A,B})_{i_t} \leftarrow (f_{A,B})_{i_t} + \Delta_t$. At the end of the stream, we are promised that $f_{A,B}$ is a valid encoding of two multi-sets $A, B \subset \{0,1\}^d$ with $|A| = |B| = s$. A two-pass streaming algorithm is allowed to make two passes over the stream before outputting an estimate.

Corollary F.1. *For $d, s \in \mathbb{N}$, there exists a 2-pass turnstile streaming algorithm which, on a stream vector $f_{A,B}$ encoding multi-sets $A, B \subset \{0,1\}^d$ with $|A| = |B| = s$, the algorithm then computes an approximate $\widehat{\mathcal{I}}$ to $\text{EMD}(A, B)$ with*

$$\text{EMD}(A, B) \leq \widehat{\mathcal{I}} \leq \tilde{O}(\log s)\, \text{EMD}(A, B)$$

with probability at least $3/4$, and uses $O(d \log d) + \text{polylog}(s, d)$ bits of space. Moreover, the algorithm stores its own randomness (i.e., does not required the random oracle model).

Proof: The only step remaining is to derandomize the algorithm (i.e., modify the algorithm so that it can store its randomness in small space). First note that the universe reduction step requires the generation of two hash functions h_i, h_{i-1} mapping a universe of size at most 2^d to a universe of size s. Moreover, for the proof of Proposition 5.1, all that is needed is 2-wise independence, since the proof only argues about the probability of a collision of a fixed pair and applies a union bound. Since a 2-wise independent hash function $h : U_1 \to U_2$ can be stored in $O(\log(|U_1| + |U_2|))$ bits of space, this only adds an additive $O(d)$ bits to the space complexity of the algorithm.

Next, note that the linear sketching algorithms of [Ind06b] and [JW18] used in the first and second passes are both are already derandomized in $\text{poly}(\log s)$-bits of space. Thus, it will suffice to consider the randomness needed to store the Quadtree T. By Remark 4, in each depth $t \in [h]$ of the Quadtree we can sample the same set $(\boldsymbol{i}_1, \boldsymbol{i}_2, \ldots, \boldsymbol{i}_{2^t}) \sim [d]$ of coordinates for each vertex at that depth (instead of independently sampling coordinates in every vertex at the same depth). Since $h = \log 2d$, the total number of bits we must sample to define an entire quadtree is $2d \log d$. Thus, after sampling such a quadtree T with $O(d \log d)$ bits, and storing these bits, the remainder of the algorithm is already stores its randomness. Moreover, because there are at most $\text{poly}(s)$ updates to $f_{A,B}$, the coordinates of each linear sketch can be stored in $O(\log s)$ bits of space, which completes the proof.

■

References

[ABIW09] Alexandr Andoni, Khanh Do Ba, Piotr Indyk, and David Woodruff. Efficient sketches for earth-mover distance, with applications. In *Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science (FOCS '2009)*, 2009.

[AIK08] Alexandr Andoni, Piotr Indyk, and Robert Krauthgamer. Earth mover distance over high-dimensional spaces. In *Proceedings of the 19th ACM-SIAM Symposium on Discrete Algorithms (SODA '2008)*, pages 343–352, 2008.

[AKO10] Alexandr Andoni, Robert Krauthgamer, and Krzysztof Onak. Streaming algorithms from precision sampling. *arXiv preprint arXiv:1011.1263*, 2010.

[AMO93] Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, 1993.

[ANOY14] Alexandr Andoni, Aleksandar Nikolov, Krzysztof Onak, and Grigory Yaroslavtsev. Parallel algorithms for geometric graph problems. In *Proceedings of the 46th ACM Symposium on the Theory of Computing (STOC '2014)*, 2014.

[AS14] Pankaj K. Agarwal and R. Sharathkumar. Approximation algorithms for bipartite matching with metric and geometric costs. In *Proceedings of the 46th ACM Symposium on the Theory of Computing (STOC '2014)*, pages 555–564, 2014.

[AWR17] Jason Altschuler, Jonathan Weed, and Philippe Rigollet. Near-linear time approximation algorithms for optimal transport via sinkhorn iteration. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS '2017)*, 2017.

[Bar96] Yair Bartal. Probabilistic approximation of metric spaces and its algorithmic applications. In *Proceedings of the 37th Annual IEEE Symposium on Foundations of Computer Science (FOCS '1996)*, 1996.

[Bar98] Yair Bartal. On approximating arbitrary metrices by tree metrics. In *Proceedings of the 30th ACM Symposium on the Theory of Computing (STOC '1998)*, 1998.

[BDI+20] Arturs Backurs, Yihe Dong, Piotr Indyk, Ilya Razenshteyn, and Tal Wagner. Scalable nearest neighbor search for optimal transport. In *Proceedings of the 37th International Conference on Machine Learning (ICLM '2020)*, 2020.

[BI14] Arturs Bačkurs and Piotr Indyk. Better embeddings for planar earth-mover distance over sparse sets. In *Proceedings of the 41st International Colloquium on Automata, Languages and Programming (ICALP '2014)*, 2014.

[CCFC02] Moses Charikar, Kevin Chen, and Martin Farach-Colton. Finding frequent items in data streams. *Automata, languages and programming*, pages 784–784, 2002.

[CCG+98] Moses Charikar, Chandra Chekuri, Ashish Goel, Sudipto Guha, and Serge Plotkin. Approximating a finite metric by a small number of tree metrics. In *Proceedings of the 39th Annual IEEE Symposium on Foundations of Computer Science (FOCS '1998)*, 1998.

[Cha02] Moses Charikar. Similarity estimation techniques from rounding algorithms. In *Proceedings of the 34th ACM Symposium on the Theory of Computing (STOC '2002)*, pages 380–388, 2002.

[Cut13] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Proceedings of Advances in Neural Information Processing Systems (NIPS '2013)*, 2013.

[EK72] Jack Edmonds and Richard M. Karp. Theoretical improvements in algorithmic efficiency for network flow problems. *Journal of the ACM*, 19(2):248–264, 1972.

[FRT04] Jittat Fakcharoenphol, Satish Rao, and Kunal Talwar. A tight bound on approximating arbitrary metrics by tree metrics. *Journal of Computer and System Sciences*, 69(3):485–497, 2004.

[GCPB16] Aude Genevay, Marco Cuturi, Gabriel Peyré, and Francis Bach. Stochastic optimization for large-scale optimal transport. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS '2016)*, 2016.

[HPIS13] Sariel Har-Peled, Piotr Indyk, and Anastasios Sidiropoulos. Euclidean spanners in high dimensions. In *Proceedings of the 24th ACM-SIAM Symposium on Discrete Algorithms (SODA '2013)*, 2013.

[Ind06a] Piotr Indyk. Stable distributions, pseudorandom generators, embeddings, and data stream computation. *Journal of the ACM*, 53(3):307–323, 2006.

[Ind06b] Piotr Indyk. Stable distributions, pseudorandom generators, embeddings, and data stream computation. *Journal of the ACM (JACM)*, 53(3):307–323, 2006.

[Ind07] Piotr Indyk. A near linear time constant factor approximation for euclidean bichromatic matching (cost). In *Proceedings of the 18th ACM-SIAM Symposium on Discrete Algorithms (SODA '2007)*, 2007.

[IT03] Piotr Indyk and Nitin Thaper. Fast color image retrieval via embeddings. In *Workshop on Statistical and Computational Theories of Vision (at ICCV)*, 2003.

[JST11] Hossein Jowhari, Mert Sağlam, and Gábor Tardos. Tight bounds for lp samplers, finding duplicates in streams, and related problems. In *Proceedings of the Thirtieth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, PODS '11, pages 49–58, New York, NY, USA, 2011. ACM.

[JW18] Rajesh Jayaram and David Woodruff. Perfect lp sampling in a data stream. In *Proceedings of the 59th Annual IEEE Symposium on Foundations of Computer Science (FOCS '2018)*, 2018.

[JW19] Rajesh Jayaram and David P Woodruff. Towards optimal moment estimation in streaming and distributed models. *arXiv preprint arXiv:1907.05816*, 2019.

[KNP19] Andrey Boris Khesin, Aleksandar Nikolov, and Dmitry Paramonov. Preconditioning for the geometric transportation problem. In *Proceedings of the 35th International Symposium on Computational Geometry (SoCG '2019)*, 2019.

[KNW10] Daniel M Kane, Jelani Nelson, and David P Woodruff. On the exact space complexity of sketching and streaming small norms. In *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms*, pages 1161–1178. SIAM, 2010.

[KSKW15] Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. From word embeddings to document distances. In *Proceedings of the 32nd International Conference on Machine Learning (ICML '2015)*, 2015.

[Kuh55] Harold W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1–2):83–97, 1955.

[LYFC19] Tam Le, Makoto Yamada, Kenji Fukumizu, and Marco Cuturi. Tree-sliced variants of wasserstein distances. In *Proceedings of Advances in Neural Information Processing Systems 31 (NeurIPS '2019)*, 2019.

[MSC+13] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of Advances in Neural Information Processing Systems (NIPS '2013)*, pages 3111–3119, 2013.

[Mun57] James Munkres. Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial and Applied Mathematics*, 5(1):32–38, 1957.

[Nag06] HN Nagaraja. Order statistics from independent exponential random variables and the sum of the top order statistics. *Advances in Distribution Theory, Order Statistics, and Inference*, pages 173–185, 2006.

[Nis92] Noam Nisan. Pseudorandom generators for space-bounded computation. *Combinatorica*, 12(4):449–461, 1992.

[PC19] Gabriel Peyré and Marco Cuturi. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5–6):355–607, 2019.

[PSM14] Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP '2014)*, pages 1532–1543, 2014.

[RTG00] Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas. The Earth Mover's Distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, 2000.

[Sam84] Hanan Samet. The quadtree and related hierarchical data structures. *ACM Computing Surveys (CSUR)*, 12(2):187–260, 1984.

[She17] Jonah Sherman. Generalized preconditioning and undirected minimum cost flow. In *Proceedings of the 28th ACM-SIAM Symposium on Discrete Algorithms (SODA '2017)*, 2017.

[YO14] Arman Yousefi and Rafail Ostrovsky. Improved approximation algorithms for earth-mover distance in data streams. *arXiv preprint arXiv:1404.6287*, 2014.