

# Fast Cross-Polytope Locality-Sensitive Hashing

Christopher Kennedy<sup>\*1</sup> and Rachel Ward<sup>†2</sup>

1 Department of Mathematics, University of Texas at Austin, USA  
ckennedy@math.utexas.edu

2 Department of Mathematics, University of Texas at Austin, USA  
rward@math.utexas.edu

---

## Abstract

We provide a variant of cross-polytope locality sensitive hashing with respect to angular distance which is provably optimal in asymptotic sensitivity and enjoys  $\mathcal{O}(d \ln d)$  hash computation time. Building on a recent result in [4], we show that optimal asymptotic sensitivity for cross-polytope LSH is retained even when the dense Gaussian matrix is replaced by a fast Johnson-Lindenstrauss transform followed by discrete pseudo-rotation, reducing the hash computation time from  $\mathcal{O}(d^2)$  to  $\mathcal{O}(d \ln d)$ . Moreover, our scheme achieves the optimal *rate of convergence* for sensitivity. By incorporating a low-randomness Johnson-Lindenstrauss transform, our scheme can be modified to require only  $\mathcal{O}(\ln^9(d))$  random bits.

**1998 ACM Subject Classification** F.2.2 Geometrical problems and computations

**Keywords and phrases** Locality-sensitive hashing, Dimension reduction, Johnson-Lindenstrauss transform

**Digital Object Identifier** 10.4230/LIPIcs.ITCS.2017.53

## 1 Introduction

The nearest neighbor search problem is an essential algorithmic component to a wide variety of applications including data compression, information retrieval, image storage, computer vision, and pattern recognition. **Nearest neighbor search (NN)** can be stated as follows: given a metric space  $(X, \mathcal{D})$  and a set of points  $P = \{x_1, \dots, x_n\} \subset X$ , for a query point  $x \in P$  find  $y = \operatorname{argmin}_{x_i \in P \setminus \{x\}} \mathcal{D}(x_i, x)$ . In high dimensions, it is known that existing algorithms have poor performance (see [23]); that is, for a query point  $x \in P$ , any algorithm for NN must essentially compute the distances between  $x$  and each point in  $P \setminus \{x\}$ .

In order to improve on linear search, one may relax the problem to that of *approximate* nearest neighbors search. Precisely, the  $(R, c)$  **near neighbor problem** ( $(R, c)$ -NN) as introduced in [12] is as follows: given a query point  $x \in P$  and the assurance of a point  $y' \in P$  such that  $\mathcal{D}(y', x) < R$ , find  $y \in P$  such that  $\mathcal{D}(y, x) < cR$ . In contrast to exact nearest neighbors search, the approximate nearest neighbor search problem can be solved in *sublinear* query time, and this is achieved using **locality sensitive hashing** (LSH). The idea in LSH is to specify a function from the domain  $X$  to a discrete set of hash values – a so-called *hash function* – which sends closer points to the same *hash value* with higher probability than points which are far apart. Then, for a set of points  $P = \{x_1, \dots, x_n\} \subset X$  and a query point  $x \in P$ , search within its corresponding hash bucket for a nearest neighbor.

---

\* C. Kennedy was supported in part by R. Ward's NSF CAREER grant and an ASOFR Young Investigator Award.

† R. Ward was supported in part by an NSF CAREER grant and an ASOFR Young Investigator Award.



From here on out, we fix the space  $X = S^{d-1}$  endowed with the euclidean metric. We begin by recalling the standard notion of sensitivity for a hash family; intuitively, a hash family with higher sensitivity is much more likely to hash points that are close to the same hash value, and thus be a better candidate for locality sensitive hashing.

► **Definition 1.** For  $r_1 \leq r_2$  and  $p_2 \leq p_1$ , a hash family  $\mathcal{H}$  is  $(r_1, r_2, p_1, p_2)$ -sensitive if for all  $x, y \in S^{d-1}$ ,

- If  $\|x - y\|_2 \leq r_1$ , then  $\Pr_{\mathcal{H}}[h(x) = h(y)] \geq p_1$ .
- If  $\|x - y\|_2 \geq r_2$ , then  $\Pr_{\mathcal{H}}[h(x) = h(y)] \leq p_2$ .

We primarily care about the case where  $r_1 = R$ ,  $r_2 = cR$ , and to quantify sensitivity of a certain scheme, we study the parameter

$$\rho = \frac{\ln(p_1^{-1})}{\ln(p_2^{-1})}. \quad (1)$$

The key result linking the sensitivity of a hash family to its performance for  $(R, c) - NN$  search is the following:<sup>1</sup>

► **Proposition 2** (Theorem 5 in [12]). *Given an  $(R, cR, p_1, p_2)$ -sensitive hash family  $\mathcal{H}$ , there exists a data structure that solves  $(R, c) - NN$  with constant probability using  $\mathcal{O}(dn + n^{1+\rho})$  space,  $\mathcal{O}(n^\rho)$  query time, and  $\mathcal{O}(n^\rho \ln_{1/p_1} n)$  evaluations of hash functions from  $\mathcal{H}$ .*

Since the parameter  $\rho$  quantifies the performance of a given LSH algorithm for  $(R, c) - NN$ , it is of interest to make this parameter as small as possible. It was shown in [20] that  $\rho = \frac{1}{c^2}$  is asymptotically (in  $d$ ) optimal for the case of unit sphere with the euclidean metric. Spherical LSH ([5], [6]) was shown to achieve this optimal sensitivity; however, the corresponding hash functions in spherical LSH are not practical to compute. Subsequently, Andoni, Indyk, Laarhoven, and Razenshteyn [4] showed the existence of an LSH scheme with optimally sensitive hash functions which are practical to implement; namely, the *cross-polytope* LSH scheme which has been previously proposed in [22] (see also [7], [20], [19]). Given a matrix  $\mathcal{G} \in \mathbb{R}^{d \times d}$  with i.i.d.  $\mathcal{N}(0, 1)$  entries, the cross polytope hash of a point  $x \in S^{d-1}$  is defined as

$$h(x) = \operatorname{argmin}_{u \in \{\pm e_i\}} \left\| \frac{\mathcal{G}x}{\|\mathcal{G}x\|_2} - u \right\|_2, \quad (2)$$

where  $\{e_i\}_{i=1}^d$  is the standard basis for  $\mathbb{R}^d$ . The paper [4] provided the following collision probability for cross-polytope LSH.

► **Proposition 3** (Theorem 1 in [4]). *Suppose  $x, y \in S^{d-1}$  are such that  $\|x - y\|_2 = R$ , with  $0 < R < 2$ , and  $\mathcal{H}$  is the hash family defined in (2). Then,*

$$\ln \left( \frac{1}{\Pr_{\mathcal{H}}[h(x) = h(y)]} \right) = \frac{R^2}{4 - R^2} \ln d + \mathcal{O}_R(\ln(\ln d)). \quad (3)$$

Consequently,

$$\rho = \frac{1}{c^2} \frac{4 - c^2 R^2}{4 - R^2} + o(1),$$

<sup>1</sup> In particular, the algorithm stores  $L$  hash tables from the family  $\mathcal{G}$ , where each  $g \sim \mathcal{G}$  is given by  $g(x) = (h_1(x), \dots, h_k(x))$ , and  $h_i \sim \mathcal{H}, i = 1 \dots k$ . Then, given a query point  $x \in S^{d-1}$ , the algorithm looks for collisions in the buckets  $g_1(x), \dots, g_L(x)$ . The choice of parameters  $k = n^\rho$ ,  $L = \ln_{1/p_1} n$  ensure that the algorithm solves  $(R, c) - NN$  with constant probability.

where here and in the sequel,  $o(1)$  means a parameter that goes to 0 as  $d \rightarrow \infty$ . This implies that the above scheme is asymptotically optimal with respect to  $\rho$ .<sup>2</sup> Still, this scheme is limited in efficiency by the  $\mathcal{O}(d^2)$  computation required to compute a dense matrix-vector multiplication in (2). To reduce this computation, [4] proposed to use a pseudo-random rotation in place of a dense Gaussian matrix, namely,

$$h(x) = \operatorname{argmin}_{u=\{\pm e_i\}} \|HD_bHD_{b'}HD_{b''}x - u\|_2, \quad (4)$$

where  $H \in \mathbb{R}^{d \times d}$  is a Hadamard matrix and  $D_b, D_{b'}, D_{b''} \in \mathbb{R}^{d \times d}$  are independent diagonal matrices with i.i.d. Rademacher entries on the diagonal. This scheme has the advantage of computing hash functions in time  $\mathcal{O}(d \ln d)$ , and was shown in [4] to *empirically* exhibit similar collision probabilities to cross-polytope LSH, but provable guarantees on the asymptotic sensitivity of this fast variant of the standard cross-polytope LSH remain open.

## 1.1 Our Contributions

### 1.1.1 Fast cross-polytope LSH with optimal asymptotic sensitivity

While we do not prove theoretical guarantees regarding the asymptotic sensitivity of the particular fast variant (4), we construct a different variant of the standard cross-polytope LSH (defined below in (5)) which also enjoys  $\mathcal{O}(d \ln d)$  matrix-vector multiplication, and for which we are able to prove optimal asymptotic sensitivity  $\rho = \frac{1}{c^2}$ :

$$h_F(x) = \operatorname{argmin}_{u=\{\pm e_i\}} \left\| \frac{\mathcal{G}(H_S D_b x)}{\|\mathcal{G}(H_S D_b x)\|_2} - u \right\|_2; \quad (5)$$

Here,  $D_b : \mathbb{R}^d \rightarrow \mathbb{R}^d$  is a diagonal matrix with i.i.d. Rademacher entries on the diagonal,  $H_S \in \mathbb{R}^{m \times d}$  is a partial Hadamard matrix restricted to a random subset  $S \subset [d]$  of  $|S| = m = \mathcal{O}(\log(d))$  rows, and  $\mathcal{G} : \mathbb{R}^m \rightarrow \mathbb{R}^{d'}$  is a Gaussian matrix that lifts and rotates in dimension  $d'$  in the range  $m \leq d' \leq d$ . There is nothing special about lifting to dimension  $d$ , and indeed one could lift to dimension  $d' > d$ , but if  $d'$  grows faster than  $d$ , the hash computation no longer takes time  $\mathcal{O}(d \ln d)$ .

The embedding  $H_S D_b x$  acts as a Johnson-Lindenstrauss (JL) transform<sup>3</sup>, and embeds the points in dimension  $m \approx \ln d$ .

It is straightforward that the hash computation  $x \rightarrow h_F(x)$  takes  $\mathcal{O}(d'm)$  time from the Gaussian matrix multiplication and  $\mathcal{O}(d \ln d)$  time from the JL transform. We will show that optimal asymptotic sensitivity is still achieved without lifting,  $d' = m$ , but we observe both empirically and theoretically that the *rate of convergence* to the asymptotic sensitivity improves by lifting to higher dimension; taking  $d'$  closer to  $d$  results in empirically closer results to the standard cross-polytope scheme (see section 5 for more details). Moreover, our scheme achieves the lower bound given by Theorem 2 in [4] for the fastest rate of convergence among all hash families which has to  $d'$  values.

<sup>2</sup> In fact, the coefficient  $\frac{4-c^2R^2}{4-R^2} < 1$  for every choice of  $c > 1$  and  $0 < R < 2$ , but this does not break the lower bound given in [20] since the lower bound  $\rho = \frac{1}{c^2}$  only holds for a particular sequence  $R = R(d)$ . For cross-polytope LSH and the schemes proposed here, any sequence  $R(d) \rightarrow 0$  suffices.

<sup>3</sup> Formally, given a finite metric space  $(X, \|\cdot\|) \subset \mathbb{R}^d$ , a JL transform is a linear map  $\Phi : \mathbb{R}^d \rightarrow \mathbb{R}^m$  such that for all  $x \in X$ ,  $(1 - \delta)\|x\|^2 \leq \|\Phi x\|^2 \leq (1 + \delta)\|x\|^2$ , with  $m \ll d$  close to the optimal scaling  $m = C\delta^{-2} \ln(|X|)$  [13, 2, 16].

■ **Table 1** Various LSH Families and corresponding Hash Functions.

LSH Family	Hash Function
Cross-Polytope LSH	$h(x) = \operatorname{argmin}_{u=\{\pm e_i\}} \left\  \frac{\mathcal{G}x}{\ \mathcal{G}x\ _2} - u \right\ _2, \quad \mathcal{G} \in \mathbb{R}^{d \times d}$
<b>Fast Cross-Polytope LSH</b>	$h_F(x) = \operatorname{argmin}_{u=\{\pm e_i\}} \left\  \frac{\mathcal{G}(H_S D_b x)}{\ \mathcal{G}(H_S D_b x)\ _2} - u \right\ _2, \quad \mathcal{G} \in \mathbb{R}^{d' \times m}$
<b>Fast Discrete Cross-Polytope LSH</b>	$h_D(x) = \operatorname{argmin}_{u=\{\pm e_i\}} \left\  \frac{\widehat{\mathcal{G}}(Zx)}{\ \widehat{\mathcal{G}}(Zx)\ _2} - u \right\ _2, \quad \widehat{\mathcal{G}} \in \mathbb{R}^{d' \times m}$

### 1.1.2 Fast cross-polytope LSH with optimal asymptotic sensitivity and few random bits

Aiming to construct a hash family with similar guarantees which also uses as little randomness as possible, we also consider a discretized version of the fast hashing scheme (5) in which the Gaussian matrix  $\mathcal{G} \in \mathbb{R}^{d' \times m}$  is replaced by a matrix  $\widehat{\mathcal{G}} \in \mathbb{R}^{d' \times m}$  whose entries are i.i.d. discrete approximations of a Gaussian; in place of the “standard” fast JL transform  $H_S D_b$ , we consider  $Z \in \mathbb{R}^{d \times m}$  a low-randomness JL transform that we will clarify later. Then, the discrete fast hashing scheme we consider is

$$h_D(x) = \operatorname{argmin}_{u=\{\pm e_i\}} \left\| \frac{\widehat{\mathcal{G}}(Zx)}{\|\widehat{\mathcal{G}}(Zx)\|_2} - u \right\|_2. \quad (6)$$

Also for this scheme, the hash computation  $x \rightarrow h(x)$  takes  $\mathcal{O}(d'm)$  time from the Gaussian matrix multiplication and  $\mathcal{O}(d \ln d)$  time from the JL transform. Our scheme has several advantages, due to the fact that the choice of  $d'$  in the range  $d \leq d' \leq m$  is flexible: To summarize our main contributions, we prove for both the fast cross-polytope LSH and the fast discrete cross-polytope LSH,

- For each  $d'$  in the range  $m \leq d' \leq d$ , this scheme achieves the asymptotically optimal  $\rho$ . Moreover, for  $d' = d$ , the rate of convergence to this  $\rho$  is optimal over all hash families with  $d$  hash values.
- With the choice  $d' = d$ , the scheme computes hashes in time  $\mathcal{O}(d \ln d)$  and performs well empirically compared to the standard cross-polytope with dense Gaussian matrix (see section 5).
- With the choice  $d' = m$ , and by discretizing the Gaussian matrix, we arrive at a scheme that has only  $\mathcal{O}(\ln^9(d))$  bits of randomness and still has optimal asymptotic sensitivity.

Table 1 contains the construction of the original cross-polytope LSH scheme, our fast cross-polytope scheme, as well as the discretized version.

## 1.2 Related work

Many of our results hinge on the careful analysis of collision probabilities for the cross-polytope LSH scheme given in [4]. Additionally, various ways to reduce the runtime of cross-polytope LSH, specifically using fast, structured projection matrices, are mentioned

in [8]. They also define a generalization of cross-polytope lsh that first projects to a low dimensional subspace, but they never consider lifting back up to a high dimensional subspace again. Johnson-Lindenstrauss transforms have previously been used in many approximate nearest neighbors algorithms, (see [12], [18], [1], [21], [5], and [9], to name a few), primarily as a preprocessing step to speed up computations that have some dependence on the dimension. LSH with p-stable distributions, as introduced in [10], uses a random projection onto a single dimension, which is later generalized in [3] to random projection onto  $o(\ln d)$  dimensions, with the latter having optimal exponent  $\rho = \frac{1}{c^2} + \mathcal{O}(\ln(\ln d)/\ln^{1/3} d)$ . We make a note that our scheme uses dimension reduction slightly differently, as an intermediate step before lifting the vectors back up to a different dimension.

Similar dimension reduction techniques have been used in [17], where the data is sparsified and then a random projection matrix is applied. The authors exploit the fact that the random projection matrix will have the restricted isometry property, which preserves pairwise distances between any two sparse vectors. This result is notable in that the reduced dimension has no dependence on  $n$ , the number of points. See section 4 for more discussion.

## 2 Notation

We now establish notation that will be used in the remainder.  $\mathcal{O}_R(f(d))$  is to mean  $\mathcal{O}_R(f(d)) = \mathcal{O}(f(d)g(R))$  for some finite valued function  $g : (0, 2) \rightarrow \mathbb{R}$ . The expression  $o(1)$  is a quantity such that  $\lim_{d \rightarrow \infty} o(1) = 0$ .  $H \in \mathbb{R}^{d \times d}$  is the Hadamard matrix.  $D_b \in \mathbb{R}^{d \times d}$  is a diagonal matrix whose entries are i.i.d. Rademacher variables. For a matrix  $M \in \mathbb{R}^{d \times d}$ ,  $M_S$  will denote the restriction of  $M$  to its rows indexed by the set  $S \subset \{1, \dots, d\}$ . The variable  $\mathcal{G}$  will always denote a matrix with i.i.d. standard normal Gaussian entries, where the matrix may vary in size. The variable  $\hat{\mathcal{G}}$  will always denote a matrix with i.i.d. copies of a discrete random variable  $X$  which roughly models a Gaussian.  $C$  will denote various constants that are bounded independent of the dimension. We will use  $m$  to denote the projected dimension of our points, where  $m \ll d$ , and  $d'$  the lifted dimension, where  $m \leq d' \leq d$ . For a vector  $x \in S^{d-1}$  we will denote  $\tilde{x} = H_S D_b x$ .

## 3 Main Results

We now formalize the intuition about how our scheme behaves relative to cross-polytope LSH.

► **Theorem 4.** *Suppose  $\mathcal{H}$  is the family of hash functions defined in (5) with the choice  $m = \mathcal{O}(\ln^5(d) \ln^4(\ln d))$ , and  $\rho$  is as defined in (1) for this particular family. Then we have (i-)*

$$\rho = \frac{1}{c^2} \frac{4 - c^2 R^2}{4 - R^2} + o(1).$$

and this hashing scheme runs in time  $\mathcal{O}(d \ln d)$ .

Moreover, we have the optimal rate of convergence,

(ii-)

$$\rho = \frac{1}{c^2} \frac{4 - c^2 R^2}{4 - R^2} + \mathcal{O}\left(\frac{1}{\ln d'}\right).$$

The lower bound given by Theorem 2 in [4] verifies the above rate of convergence is in fact optimal. We remark that when hashing  $n$  points simultaneously, the embedded dimension  $m$

picks up a factor of  $\ln(n)$ . Assuming that  $n$  is polynomial in  $d$ , the result in Theorem 4 still holds simultaneously over all pairs of points.

In addition to creating a fast hashing scheme, one can reduce the amount of randomness involved. In particular, we show that a slight alteration of the scheme still achieves the optimal  $\rho$ -value while using only  $\mathcal{O}(\ln^9 d)$  bits of randomness. The idea is to replace the Gaussian matrix by a matrix of i.i.d. discrete random variables. Some care is required in tuning the size of this matrix so that the correct number of bits is achieved. As a consequence the number of hash values for this scheme is of order  $\mathcal{O}(m)$  (i.e. we lift up to a smaller dimension), which lowers performance in practice, but does not affect the asymptotic sensitivity  $\rho$ . We additionally use a JL transform developed by Kane and Nelson [14] that only uses  $\mathcal{O}(\ln(d) \ln(\ln d))$  bits of randomness. Specifically, the hash function for this scheme is

$$h_D(x) = \operatorname{argmin}_{u=\{\pm e_i\}} \left\| \frac{\widehat{\mathcal{G}}(Zx)}{\|\widehat{\mathcal{G}}(Zx)\|_2} - u \right\|_2$$

where  $\widehat{\mathcal{G}} \in \mathbb{R}^{d' \times m}$  is a matrix with i.i.d. copies of a discrete random variable  $X$  which roughly models a Gaussian, and  $Z \in \mathbb{R}^{d \times m}$  is the JL transform constructed in [14]. Our analysis allows us to pick the threshold value  $d' = m$  to minimize the number of random bits.

► **Theorem 5.** *There is a hash family  $\mathcal{H}$  with  $\mathcal{O}(\ln^9 d)$  bits of randomness that achieves the bound*

$$\rho = \frac{1}{c^2} \frac{4 - c^2 R^2}{4 - R^2} + o(1),$$

and runs in time  $\mathcal{O}(d \ln d)$ .

### 3.1 Theorem 4 Part (i-) Proof Outline

First we state an elementary limit result that we will apply to the proofs of both Theorem 4 and Theorem 5.

► **Lemma 6.** *Suppose  $m_d(a), m_d(b)$  are positive functions,  $\lim_{d \rightarrow \infty} m_d(a) = a$ ,  $\lim_{d \rightarrow \infty} m_d(b) = b$ , and that  $f(d), g(d)$  are also positive,  $\lim_{d \rightarrow \infty} f(d) = \lim_{d \rightarrow \infty} g(d) = \infty$ ,  $\lim_{d \rightarrow \infty} \frac{f(d)}{g(d)} = \infty$ . Then,*

$$\lim_{d \rightarrow \infty} \frac{m_d(a)f(d) + g(d)}{m_d(b)f(d) + g(d)} = \frac{a}{b}$$

Proceeding to the proof of Theorem 4, the key observation is that for  $x, y \in S^{d-1}$ ,  $\mathcal{G}\tilde{x} = \mathcal{G}_0 \begin{bmatrix} \tilde{x} \\ 0 \end{bmatrix}$ , where  $\mathcal{G}_0 \in \mathbb{R}^{d' \times d'}$  is a square Gaussian matrix. Thus,

$$\Pr[h_f(x) = h_f(y)] = \Pr \left[ h \left( \begin{bmatrix} \tilde{x} \\ 0 \end{bmatrix} \right) = h \left( \begin{bmatrix} \tilde{y} \\ 0 \end{bmatrix} \right) \right],$$

recalling that  $h_f$  is the fast cross-polytope hash function and  $h$  is the standard version. It then follows that, provided the distance between  $\tilde{x}$  and  $\tilde{y}$  is close to the distance between  $x$  and  $y$ , we can apply proposition 3 to control the above probability. We start with a lemma for our chosen JL transform that combines a recent improvement on the *restricted isometry property* (RIP) for partial Hadamard matrices [11] with a reduction from RIP to Johnson-Lindenstrauss transforms in [15]; we defer the proof to the appendix.

► **Lemma 7.** *Suppose  $\gamma > 0$ ,  $x, y \in S^{d-1}$ ,  $\tilde{x} = H_S D_b x$ ,  $\tilde{y} = H_S D_b y$  and  $H_S \in \mathbb{R}^{m \times d}$  is such that  $m = \mathcal{O}(\gamma \ln^4(d) \ln^4(\ln d))$ . Then with probability  $1 - \mathcal{O}(d^{-\gamma})$ ,*

$$\left(1 - \frac{1}{\ln d}\right) \leq \|\tilde{x}\|_2^2 \leq \left(1 + \frac{1}{\ln d}\right), \quad (7)$$

$$\left(1 - \frac{1}{\ln d}\right) \leq \|\tilde{y}\|_2^2 \leq \left(1 + \frac{1}{\ln d}\right), \quad (8)$$

$$\left(1 - \frac{1}{\ln d}\right) \|x - y\|_2^2 \leq \|\tilde{x} - \tilde{y}\|_2^2 \leq \left(1 + \frac{1}{\ln d}\right) \|x - y\|_2^2 \quad (9)$$

We apply the above lemma with the choice  $\gamma = \ln d$  to get that

$$\frac{\|x - y\|_2^2}{\left(1 - \frac{1}{\ln d}\right)} - \frac{5}{\ln d - 1} \leq \left\| \frac{\tilde{x}}{\|\tilde{x}\|_2} - \frac{\tilde{y}}{\|\tilde{y}\|_2} \right\|_2^2 \leq \frac{\|x - y\|_2^2}{\left(1 + \frac{1}{\ln d}\right)} + \frac{5}{\ln d + 1}. \quad (10)$$

with probability  $1 - \mathcal{O}(d^{-\ln d})$ . Combining this fact with proposition 3 we get that

$$\Pr[h_f(x) = h_f(y)] = C(d')^{\frac{-\tilde{R}^2}{4-\tilde{R}^2}} \ln^{-1}(d'),$$

where  $\tilde{R} = \|\tilde{x} - \tilde{y}\|^2$  (by equation (10)) goes to  $R$  as  $d \rightarrow \infty$ , and  $C$  is bounded in the dimension. We then apply lemma 6 to see that

$$\begin{aligned} \rho &= \frac{\frac{\tilde{R}^2}{4-\tilde{R}^2} \ln(d') + \ln \ln(d') + C}{\frac{c^2 \tilde{R}^2}{4-c^2 \tilde{R}^2} \ln(d') + \ln \ln(d') + C} \\ &= \frac{1}{c^2} \frac{4 - c^2 \tilde{R}^2}{4 - \tilde{R}^2} + o(1). \end{aligned}$$

We defer the proof of Theorem 4 part (ii-) to the appendix.

### 3.2 Theorem 5 Proof Outline

We will use the following result (formulated as an analogue to lemma 7), due to Kane and Nelson, that reduces the amount of randomness required to perform a JL transform.

► **Proposition 8.** *(Theorem 13 and Remark 14 in [14]) Suppose  $\gamma > 0$ ,  $x, y \in S^{d-1}$ . Then, there is a random matrix  $Z \in \mathbb{R}^{d \times m}$  with  $m = \mathcal{O}(\gamma \ln^3(d))$  and sampled with  $\mathcal{O}(\gamma \ln^2(d))$  random bits such that with probability  $1 - \mathcal{O}(d^{-\gamma})$ ,*

$$\left(1 - \frac{1}{\ln d}\right) \leq \|Zx\|_2^2 \leq \left(1 + \frac{1}{\ln d}\right),$$

$$\left(1 - \frac{1}{\ln d}\right) \leq \|Zy\|_2^2 \leq \left(1 + \frac{1}{\ln d}\right),$$

$$\left(1 - \frac{1}{\ln d}\right) \|x - y\|_2^2 \leq \|Z(x - y)\|_2^2 \leq \left(1 + \frac{1}{\ln d}\right) \|x - y\|_2^2$$

Now we want to construct a hash scheme that uses a Gaussian rotation with which to compare our discretized scheme. Define

$$h'_D(x) = \operatorname{argmin}_{u=\{\pm e_i\}} \left\| \frac{\mathcal{G}' Z x}{\|\mathcal{G}' Z x\|_2} - u \right\|_2, \quad (11)$$

where  $\mathcal{G}' \in \mathbb{R}^{m \times m}$  is a standard i.i.d. Gaussian matrix. The following elementary lemma gives us a suitable replacement for each Gaussian in the matrix  $\mathcal{G}'$ .

► **Lemma 9.** *Suppose  $g \sim \mathcal{N}(0, 1)$ . Then, there is a symmetric, discrete random variable  $X$  taking  $2^b$  values such that for any  $x \in \mathbb{R}$ ,*

$$\Pr[g \leq x] = \Pr[X \leq x] + \mathcal{O}(2^{-b}) \quad (12)$$

The discretized scheme can now be constructed by

$$h_D(x) = \operatorname{argmin}_{u=\{\pm e_i\}} \left\| \frac{\widehat{\mathcal{G}}Zx}{\|\widehat{\mathcal{G}}Zx\|_2} - u \right\|_2, \quad (13)$$

where the entries of  $\widehat{\mathcal{G}} \in \mathbb{R}^{d' \times m}$  are i.i.d. copies of the random variable  $X$  in Lemma 9. Note that each discrete random variable has  $b$  bits of randomness, so the hashing scheme has minimal randomness when  $d' = m$ , thus there are  $m \times m \times b + \mathcal{O}(\gamma \ln^2(d)) = \mathcal{O}(\gamma^2 \ln^6(d)b + \gamma \ln^2(d))$  bits of randomness. As we will see, we can choose  $\gamma$  and  $b$  to be a power of  $\ln(d)$  while still achieve the optimal asymptotic  $\rho$ . For this we have the following lemma.

► **Lemma 10.** *Let  $x, y \in \mathbb{R}^d$  be such that  $\|x - y\|_2 = R$ ,  $\tilde{x} = Zx$ , and let  $h, h'$  be as defined in (13) and (11) respectively with  $m = \mathcal{O}(\ln^4(d))$ ,  $b = \log_2(d)$  where  $\tilde{R} = \|\tilde{x} - \tilde{y}\|_2$ . Then,*

$$\ln(\Pr[h_D(x) = h_D(y)]) = \ln(\Pr[h'_D(x) = h'_D(y)]) + \mathcal{O}_{\tilde{R}}(1) \quad (14)$$

We defer the proof of lemma 10 to the appendix, but the idea is as follows. We can first write

$$\Pr[h'_D(x) = h'_D(y)] = 2d' \Pr[h'_D(x) = h'_D(y) = e_1].$$

Note that the set  $\{h'_D(x) = h'_D(y) = e_1\} = \{(\mathcal{G}'\tilde{x})_1 \geq |(\mathcal{G}'\tilde{x})_2|, (\mathcal{G}'\tilde{y})_1 \geq |(\mathcal{G}'\tilde{y})_2|\}$ , which is the Gaussian measure of a convex polytope, so we can write the above probability as the integral over  $m$  intervals of the  $m$ -dimensional Gaussian probability distribution. We can then use equation (12) to replace the Gaussian pdf with the discrete Gaussian pdf in each coordinate successively, and (keeping track of parameters), the lemma follows.

We now run the same argument as in Theorem 4 by setting  $\gamma = \ln d$ , so combining lemma 10 and proposition 3 applied to  $h'_D(x)$ , we have that

$$\begin{aligned} \rho &= \frac{\ln(\Pr[h_D(x) = h_D(y)])}{\ln(\Pr[h_D(cx) = h_D(cy)])} \\ &= \frac{\ln(\Pr[h'_D(x) = h'_D(y)]) + \mathcal{O}_{\tilde{R}}(1)}{\ln(\Pr[h'_D(cx) = h'_D(cy)]) + \mathcal{O}_{\tilde{R}}(1)} \\ &= \frac{\frac{R_+^2}{4-R_+^2} \ln(d') + \ln \ln(d') + C + \mathcal{O}_{\tilde{R}}(1)}{\frac{c^2 R_-^2}{4-c^2 R_-^2} \ln(d') + \ln \ln(d') + C + \mathcal{O}_{\tilde{R}}(1)} \\ &= \frac{\frac{R_+^2}{4-R_+^2} \ln(d') + \ln \ln(d') + C}{\frac{c^2 R_-^2}{4-c^2 R_-^2} \ln(d') + \ln \ln(d') + C} \\ &= \frac{1}{c^2} \frac{4 - c^2 R_-^2}{4 - R_+^2} + o(1), \text{ by lemma 6.} \end{aligned}$$

Finally, by our choice of  $\gamma$  and  $b$  in the above lemma, we know that there are  $\mathcal{O}(\ln^9(d))$  bits of randomness.



## 4 Open Problems

Although we achieve a logarithmic number of bits of randomness in Theorem 5, there is no reason to believe this is optimal among all hash families. More generally, given a particular rate of convergence to the optimal asymptotic sensitivity we would like to know the minimal number of required bits of randomness. Note that by the result in [20], for each dimension  $d$ ,  $c > 0$ , and  $q > 0$ , there is some distance  $R > 0$  such that the sensitivity parameter  $\rho \geq \frac{1}{c^2} - \mathcal{O}_q\left(\frac{1}{\ln d}\right)$ . In light of this result, we would like to know, for a given rate of convergence, whether it gets close to the lower bound  $\frac{1}{c^2}$  for all sequences of distances  $R = R(d)$ . Note that this condition holds for cross-polytope lsh with  $f(d) = \mathcal{O}\left(\frac{1}{\ln d}\right)$ .

► **Problem 11.** *Given a rate of convergence  $f(d)$  such that  $\lim_{d \rightarrow \infty} f(d) = 0$ , find the minimal number of bits  $\mathcal{O}_f(d)$  such that any hash family  $\mathcal{H}$  over the sphere  $S^{d-1}$  with  $\mathcal{O}_f(d)$  bits of randomness satisfies  $\rho = \frac{1}{c^2} + f(d)$  for all sequences  $R = R(d)$ .*

A more practical question is, given a rate of convergence for  $\rho$ , what is the fastest one could compute a hash family achieving this rate.

► **Problem 12.** *Given a rate of convergence  $f(d)$  as in Problem 11, find the hash family  $\mathcal{H}$  over  $S^{d-1}$  such that  $\rho = \frac{1}{c^2} + f(d)$  for all sequences  $R = R(d)$ , that also has the fastest hash computations.*

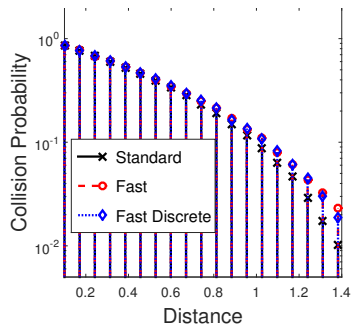
It would be natural to extend our theoretical analysis to the case of hashing a collection of  $n$  points simultaneously. In this setting, the embedding dimension of the JL matrix would inherit an additive factor depending on  $\ln(n)$ . Inspired by the construction in [17] which first sparsifies the data then exploits the restricted isometry property which applies uniformly over all sparse vectors, we can aim for a construction that doesn't depend on the number of data points.

## 5 Numerical Experiments

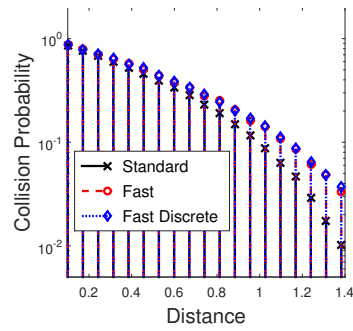
To illustrate our theoretical results in the low dimensional case, we ran Monte Carlo simulations to compare the collision probabilities for regular cross-polytope LSH as well as the fast and discrete versions for various values of the original and lifted dimension. We refer to [4] for an in depth comparison of run times for cross-polytope LSH and other popular hashing schemes.

The experiments were run with  $N = 20000$  trials. For each trial, two points were fixed at the given distance threshold on the unit sphere (in the plane given by the first two coordinates) and then rotated uniformly at random on the sphere. The discretized scheme used 10 bits of randomness for each entry. The fast, discrete, and regular cross-polytope LSH schemes exhibit similar collision probabilities for small distances, with fast/discrete cross-polytope having marginally higher collision probabilities for larger distances. It is clear that as the lifted dimension decreases, the fast and discrete versions have higher collision probabilities at further distances, which decreases the sensitivity of those schemes.

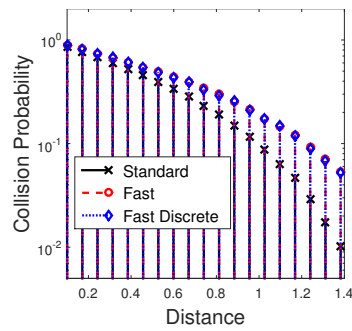
## 53:10 Fast Cross-Polytope Locality-Sensitive Hashing



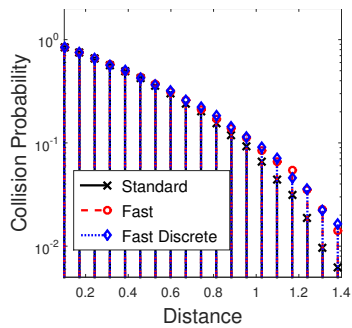
■ Figure 1  $d = 128, d' = 128$ .



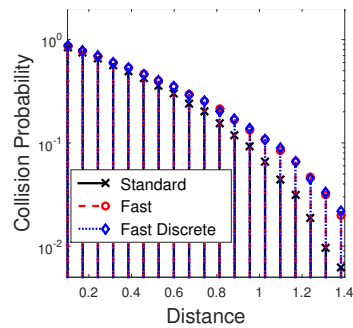
■ Figure 2  $d = 128, d' = 64$ .



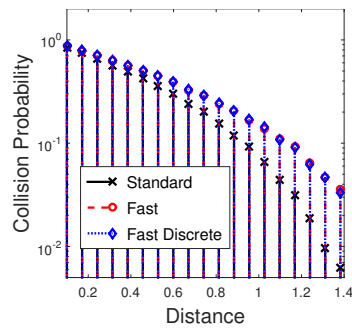
■ Figure 3  $d = 128, d' = 32$ .



■ Figure 4  $d = 256, d' = 256$ .

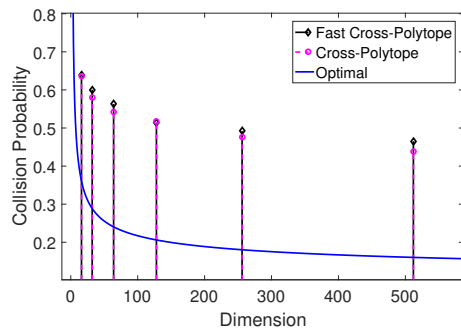


■ Figure 5  $d = 256, d' = 128$ .

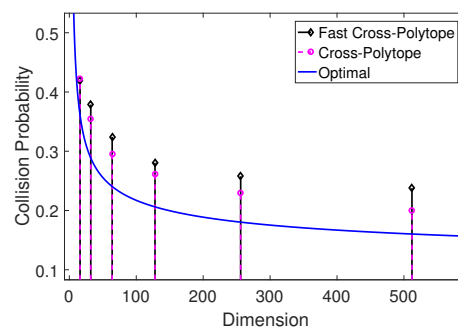


■ Figure 6  $d = 256, d' = 64$ .

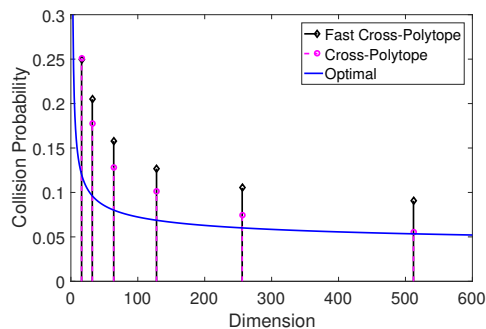
The following figures illustrate the rate of convergence to the optimal collision probability as  $d \rightarrow \infty$ , as well as various lines that illustrate the optimal rate of convergence  $C \setminus \ln(d)$ , where  $C$  varies for illustrative purposes. The experiments were run with varying distances and clearly show the same rate of convergence for the collision probability between the standard and fast cross-polytope schemes. We note that at low dimensions, the schemes behave even more similarly because the embedded dimension is much closer to the original dimension in this case.



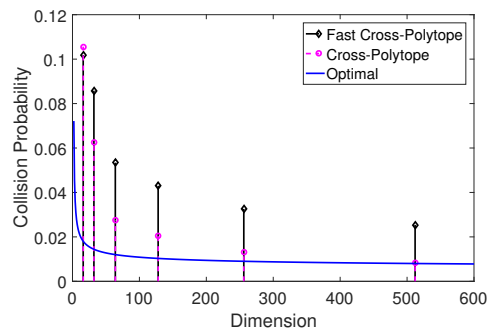
■ **Figure 7**  $R = 0.4$ .



■ **Figure 8**  $R = 0.7$ .



■ **Figure 9**  $R = 1$ .



■ **Figure 10**  $R = 1.3$ .

## References

- 1 Nir Ailon and Bernard Chazelle. The fast Johnson-Lindenstrauss transform and approximate nearest neighbors. *SIAM J. Comput.*, 39(1):302–322, May 2009. doi:10.1137/060673096.
- 2 Noga Alon. Problems and results in extremal combinatorics. *Discrete Mathematics*, 273:31–53, 2003.
- 3 Alexandr Andoni and Piotr Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science, FOCS '06*, pages 459–468, Washington, DC, USA, 2006. IEEE Computer Society. doi:10.1109/FOCS.2006.49.
- 4 Alexandr Andoni, Piotr Indyk, Thijs Laarhoven, Ilya Razenshteyn, and Ludwig Schmidt. Practical and optimal lsh for angular distance. In *Proceedings of the 28th International Conference on Neural Information Processing Systems, NIPS'15*, pages 1225–1233,

- Cambridge, MA, USA, 2015. MIT Press. URL: <http://dl.acm.org/citation.cfm?id=2969239.2969376>.
- 5 Alexandr Andoni, Piotr Indyk, Huy L Nguyen, and Ilya Razenshiteyn. Beyond Locality-Sensitive Hashing. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1018–1028. SIAM, 2014.
  - 6 Alexandr Andoni and Ilya Razenshiteyn. Optimal data-dependent hashing for approximate near neighbors. In *Proceedings of the Forty-Seventh Annual ACM Symposium on Theory of Computing*, STOC '15, pages 793–801, New York, NY, USA, 2015. ACM. doi:10.1145/2746539.2746553.
  - 7 Alexandr Andoni and Ilya Razenshiteyn. Tight Lower Bounds for Data-Dependent Locality-Sensitive Hashing. *ArXiv e-prints*, July 2015. arXiv:1507.04299.
  - 8 Anja Becker and Thijs Laarhoven. Efficient (ideal) lattice sieving using cross-polytope lsh. Cryptology ePrint Archive, Report 2015/823, 2015. URL: <http://eprint.iacr.org/>.
  - 9 Anirban Dasgupta, Ravi Kumar, and Tamás Sarlós. Fast locality-sensitive hashing. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1073–1081. ACM, 2011.
  - 10 Mayur Datar, Nicole Immorlica, Piotr Indyk, and Vahab Mirrokni. Locality sensitive hashing scheme based on p-stable distributions. In *Proceedings of the twentieth annual Symposium on Computational Geometry*. New York, pages 253–262, 2004.
  - 11 Ishay Haviv and Oded Regev. The restricted isometry property of subsampled fourier matrices. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '16, pages 288–297, Philadelphia, PA, USA, 2016. Society for Industrial and Applied Mathematics. URL: <http://dl.acm.org/citation.cfm?id=2884435.2884457>.
  - 12 Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing*, STOC '98, pages 604–613, New York, NY, USA, 1998. ACM. doi:10.1145/276698.276876.
  - 13 William B Johnson and Joram Lindenstrauss. Extensions of lipschitz mappings into a hilbert space. *Contemporary mathematics*, 26:189–206, 1984.
  - 14 Daniel M. Kane and Jelani Nelson. Sparser johnson-lindenstrauss transforms. *J. ACM*, 61(1):4:1–4:23, January 2014. doi:10.1145/2559902.
  - 15 Felix Kraemer and Rachel Ward. New and improved Johnson-Lindenstrauss embeddings via the restricted isometry property. *SIAM Journal on Mathematical Analysis*, 43(3):1269–1281, 2011.
  - 16 Kasper Green Larsen and Jelani Nelson. The johnson-lindenstrauss lemma is optimal for linear dimensionality reduction. *arXiv preprint arXiv:1411.2404*, 2014.
  - 17 Yue Lin, Rong Jin, Deng Cai, Shuicheng Yan, and Xuelong Li. Compressed hashing. In *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '13, pages 446–451, Washington, DC, USA, 2013. IEEE Computer Society. doi:10.1109/CVPR.2013.64.
  - 18 Ting Liu, Andrew W. Moore, Ke Yang, and Alexander G. Gray. An investigation of practical approximate nearest neighbor algorithms. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 825–832. MIT Press, 2005. URL: <http://papers.nips.cc/paper/2666-an-investigation-of-practical-approximate-nearest-neighbor-algorithms.pdf>.
  - 19 Rajeev Motwani, Assaf Naor, and Rina Panigrahi. Lower bounds on Locality Sensitive Hashing. In *Proceedings of the Twenty-second Annual Symposium on Computa-*

- tional Geometry*, SCG '06, pages 154–157, New York, NY, USA, 2006. ACM. doi: 10.1145/1137856.1137881.
- 20 Ryan O'Donnell, Yi Wu, and Yuan Zhou. Optimal lower bounds for Locality-Sensitive Hashing (except when  $Q$  is tiny). *ACM Trans. Comput. Theory*, 6(1):5:1–5:13, March 2014. doi:10.1145/2578221.
  - 21 Andrei Osipov. *A Randomized Approximate Nearest Neighbors Algorithm*. PhD thesis, Yale University, New Haven, CT, USA, 2011. AAI3467911.
  - 22 Kengo Terasawa and Yuzuru Tanaka. Spherical LSH for approximate nearest neighbor search on unit hypersphere. In *Algorithms and Data Structures*, pages 27–38. Springer, 2007.
  - 23 Roger Weber, Hans-Jörg Schek, and Stephen Blott. A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. In *Proceedings of the 24rd International Conference on Very Large Data Bases*, VLDB '98, pages 194–205, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.

## 6 Appendix

### 6.1 Proof of Theorem 4 Part (ii-)

Let  $\rho_{R,c}$  be the exponent for standard cross-polytope lsh in dimension  $d'$ , and  $\rho_{R,c}^{fast}$  be the exponent for fast cross-polytope lsh lifted to dimension  $d'$ . Suppose that

$$\rho_{R,c} - \frac{1}{c^2} \frac{4 - c^2 R^2}{4 - R^2} \leq C(R, c)F(d'),$$

where  $F(d') \rightarrow 0$  as  $d' \rightarrow \infty$  and  $C(r, c)$  is constant in the dimension  $d'$ . Assume that  $H_s D_b : \mathbb{R}^d \rightarrow \mathbb{R}^m$  is a  $\delta$ -isometry on  $x - y$ , i.e.

$$\|x - y\|_2^2 \leq R^2 \implies \|\tilde{x} - \tilde{y}\|_2^2 \leq (1 + \delta)R^2 \tag{15}$$

$$\|x - y\|_2^2 \geq c^2 R^2 \implies \|\tilde{x} - \tilde{y}\|_2^2 \geq (1 - \delta)c^2 R^2. \tag{16}$$

The next observation is that  $h_f(x)$  applies the standard cross-polytope lsh scheme on  $H_s D_b x$ , so conditioned on  $H_s D_b x$  being a  $\delta$ -isometry, we can analyze the fast scheme in terms of the standard scheme as follows:

$$\rho_{R,c}^{fast} \leq \rho_{R',c'},$$

where  $R' = R\sqrt{1 + \delta}$ ,  $c' = \sqrt{\frac{1-\delta}{1+\delta}}c$ . Now, we can say

$$\begin{aligned} \rho_{R,c}^{fast} - \frac{1}{c^2} \frac{4 - c^2 R^2}{4 - R^2} &\leq [\rho_{R,c}^{fast} - \rho_{R',c'}] + \left[ \rho_{R',c'} - \frac{1}{(c')^2} \frac{4 - (c')^2 (r')^2}{4 - (R')^2} \right] \\ &\quad + \left[ \frac{1}{(c')^2} \frac{4 - (c')^2 (R')^2}{4 - (R')^2} - \frac{1}{c^2} \frac{4 - c^2 R^2}{4 - R^2} \right] \\ &\leq C(R', c')F(d) + \left[ \frac{1}{(c')^2} \frac{4 - (c')^2 (R')^2}{4 - (R')^2} - \frac{1}{c^2} \frac{4 - c^2 R^2}{4 - R^2} \right]. \end{aligned}$$

The difference in the last equation can be bounded as

$$\begin{aligned}
 & \frac{1}{(c')^2} \frac{4 - (c')^2(R')^2}{4 - (R')^2} - \frac{1}{c^2} \frac{4 - c^2R^2}{4 - R^2} = \left( \frac{1 + \delta}{c^2(1 - \delta)} \right) \frac{4 - (1 - \delta)c^2R^2}{4 - (1 - \delta)R^2} - \frac{1}{c^2} \frac{4 - c^2R^2}{4 - R^2} \\
 & \leq \frac{(1 + \delta)(4 - (1 - \delta)c^2R^2)(4 - R^2) - (4 - c^2R^2)(1 - \delta)(4 - (1 - \delta)R^2)}{\frac{c^2}{2}(4 - R^2)^2} \\
 & = \delta \mathcal{O}(R, c) + \frac{(1 + \delta)(4 - c^2R^2)(4 - R^2) - (1 - \delta)(4 - c^2R^2)(4 - R^2)}{\frac{c^2}{2}(4 - R^2)^2} \\
 & = \delta D(R, c),
 \end{aligned}$$

so it follows that  $\rho_{R,c}^{fast} - \frac{1}{c^2} \frac{4 - c^2R^2}{4 - R^2} \leq \delta D(R, c) + C(R', c')F(d')$  conditioned on the fact that  $H_s D_b$  is a  $\delta$ -isometry on  $x - y$ . Note that for  $d'$  large enough,  $C(R', c')$  is bounded above by a constant independent of the dimension. We can make the choice  $\delta = \frac{1}{\ln(d')}$ , so that the isometry condition holds with probability  $1 - \mathcal{O}(d^{-\ln d})$ , so if  $\rho$  is the true exponent without conditioning, we get that

$$\begin{aligned}
 \rho & \leq \frac{p_1}{p_2 + C \ln(1 - d^{-\ln d})} \\
 & \leq \frac{p_1}{p_2 - C d^{-\ln d}} \\
 & \leq \frac{p_1}{p_2} (1 + C d^{-\ln d} / p_1),
 \end{aligned}$$

where  $C > 0$  is a constant that changes by line but is independent of the dimension. From this expression it is easy to see that the error term decays at least like  $1/\ln d'$  (recall that  $d' \leq d$ ).

Finally, provided  $F(d')$  decays as fast as  $\frac{1}{\ln(d')}$ , the result will hold. This follows from Theorem 1 in [4].

## 6.2 Proof of Lemma 6

We know that for any  $\epsilon > 0$  and  $d$  large enough,  $m_d(b) \geq b - \epsilon$ , so that

$$\begin{aligned}
 \lim_{d \rightarrow \infty} \frac{g(d)}{m_d(b)f(d) + g(d)} & \leq \lim_{d \rightarrow \infty} \frac{g(d)}{(b - \epsilon)f(d) + g(d)} \\
 & = \lim_{d \rightarrow \infty} \frac{1}{(b - \epsilon) \frac{f(d)}{g(d)} + 1} = 0,
 \end{aligned}$$

and by positivity the inequality is an equality. This implies that

$$\lim_{d \rightarrow \infty} \frac{m_d(a)f(d) + g(d)}{m_d(b)f(d) + g(d)} = \lim_{d \rightarrow \infty} \frac{m_d(a)f(d)}{m_d(b)f(d) + g(d)}.$$

The same argument on the reciprocal shows that

$$\lim_{d \rightarrow \infty} \frac{m_d(a)f(d)}{m_d(b)f(d) + g(d)} = \lim_{d \rightarrow \infty} \frac{m_d(a)f(d)}{m_d(b)f(d)} = \frac{a}{b}$$

## 6.3 Proof of Lemma 7

Define the event

$$E_{v,\delta} := \{v \in \mathbb{R}^n : (1 - \delta)\|v\|_2 \leq \|\tilde{v}\|_2 \leq (1 + \delta)\|v\|_2\}.$$

Combining Theorem 4.5 of [11] and Theorem 3.1 of [15], we know that for any  $\eta \in (0, 1)$ , any  $s \geq 40 \ln(12/\eta)$ , some  $C_0 > 0$ , and provided  $m = \mathcal{O}(\delta^{-2} \ln^2(1/\delta) s \ln^2(s/\delta) \ln(d))$ ,

$$\Pr[E_{x,\delta} \cap E_{y,\delta} \cap E_{x-y,\delta}] \geq (1 - \eta)(1 - 2^{-C_0 \ln(d) \ln(s/\delta)})$$

Setting  $\delta = 1/\ln(d)$ ,  $\eta = d^{-\gamma}$ ,  $s = 40C \ln(12d)$ , we get

$$\Pr[E_{x,\delta} \cap E_{y,\delta} \cap E_{x-y,\delta}] \geq (1 - d^{-\gamma})(1 - 2^{-C_0 \ln(d) \ln(40\gamma \ln(12d) \ln(d))}),$$

and the lemma follows.

## 6.4 Proof of Lemma 10

Note that since the entries of  $\widehat{\mathcal{G}}\tilde{x}$  are symmetric and i.i.d., the probability of hashing to one value is equal for all hash values, so we get

$$\begin{aligned} \Pr[h_D(x) = h_D(y)] &= 2d' \Pr[h_D(x) = h_D(y) = e_1] \\ &= 2d' \Pr[|\cap_{j=2}^{d'} (\widehat{\mathcal{G}}\tilde{x})_1 \geq |(\widehat{\mathcal{G}}\tilde{x})_j|, (\widehat{\mathcal{G}}\tilde{y})_1 \geq |(\widehat{\mathcal{G}}\tilde{y})_j|] \\ &= 2d' \mathbb{E}_{(\widehat{\mathcal{G}}\tilde{x})_1, (\widehat{\mathcal{G}}\tilde{y})_1} (\Pr[|(\widehat{\mathcal{G}}\tilde{x})_1 \geq |(\widehat{\mathcal{G}}\tilde{x})_2|, (\widehat{\mathcal{G}}\tilde{y})_1 \geq |(\widehat{\mathcal{G}}\tilde{y})_2|]^{d'-1}). \end{aligned} \quad (17)$$

Our goal is to bound the probability  $\Pr[|(\widehat{\mathcal{G}}\tilde{x})_1 \geq |(\widehat{\mathcal{G}}\tilde{x})_2|, (\widehat{\mathcal{G}}\tilde{y})_1 \geq |(\widehat{\mathcal{G}}\tilde{y})_2|]$  in terms of the probability  $\Pr[|(\mathcal{G}'\tilde{x})_1 \geq |(\mathcal{G}'\tilde{x})_2|, (\mathcal{G}'\tilde{y})_1 \geq |(\mathcal{G}'\tilde{y})_2|]$ . Define  $E_{\mathcal{G}'}$  =  $\{(\mathcal{G}'\tilde{x})_1 \geq |(\mathcal{G}'\tilde{x})_2|, (\mathcal{G}'\tilde{y})_1 \geq |(\mathcal{G}'\tilde{y})_2|\}$  and similarly for  $\widehat{\mathcal{G}}$ . Since  $E_{\mathcal{G}'}$  is a convex polytope, we can write

$$\Pr[E_{\mathcal{G}'}] = \int_{I_1} \int_{I_2(x_1)} \dots \int_{I_m(x_1, x_2, \dots, x_{m-1})} \frac{1}{(2\pi)^m} e^{-(x_1^2 + \dots + x_m^2)/2} dx_m \dots dx_1,$$

where each  $I_j(x_1, \dots, x_j)$  is a (possibly unbounded) interval. By construction of  $X$ ,

$$\int_{I_j(x_1, \dots, x_j)} \frac{1}{2\pi} e^{-x_{j+1}^2/2} dx_{j+1} = \int_{I_j(x_1, \dots, x_j)} p_X(x_{j+1}) dx_{j+1} + \mathcal{O}(2^{-b})$$

where  $p_X(x)$  is the pdf of  $X$ . This implies that

$$\begin{aligned} \Pr[E_{\mathcal{G}'}] &= \int_{I_1} \int_{I_2(x_1)} \dots \int_{I_m(x_1, \dots, x_{m-1})} \frac{1}{(2\pi)^{m-1}} e^{-(x_1^2 + \dots + x_{m-1}^2)/2} p_X(x_m) dx_m \dots dx_1 + \mathcal{O}(2^{-b}) \\ &\dots = \int_{I_1} \int_{I_2(x_1)} \dots \int_{I_m(x_1, \dots, x_{m-1})} p_X(x_1) \dots p_X(x_m) dx_m \dots dx_1 + \mathcal{O}(m2^{-b}) \\ &= \Pr[E_{\widehat{\mathcal{G}}}] + \mathcal{O}(m2^{-b}). \end{aligned}$$

Plugging this into (17), we get

$$\begin{aligned} \Pr[h_D(x) = h_D(y)] &= 2d' \mathbb{E}_{(\widehat{\mathcal{G}}\tilde{x})_1, (\widehat{\mathcal{G}}\tilde{y})_1} (\Pr[E_{\mathcal{G}'}] + \mathcal{O}(m2^{-b}))^{d'-1} \\ &= 2d' \mathbb{E}_{(\widehat{\mathcal{G}}\tilde{x})_1, (\widehat{\mathcal{G}}\tilde{y})_1} \left[ \sum_{k=1}^{d'-1} \binom{d'-1}{k} \Pr[E_{\mathcal{G}'}]^k (\mathcal{O}(m2^{-b}))^{d'-1-k} \right]. \end{aligned}$$

We now make the choice  $m = C \ln^4(d)$ ,  $b = \log_2(d) \ln(d)$ , so that the above summation becomes

$$\begin{aligned} \sum_{k=1}^{d'-1} \binom{d'-1}{k} \Pr[E_{\mathcal{G}'}]^{d'-1-k} (C \ln^4(d) d^{-\ln(d)})^k \\ = \sum_{k=1}^{d'-1} \binom{d'-1}{k} \Pr[E_{\mathcal{G}'}]^{d'-1-k} (C \ln^4(d) d^{-\ln(d)})^k \end{aligned}$$

### 53:16 Fast Cross-Polytope Locality-Sensitive Hashing

This first term in the summation is the main term  $\Pr[E_{\mathcal{G}'}]^{d'-1}$  and the other terms can be bounded using Sterling's approximation as follows,

$$\binom{d'-1}{k} \Pr[E_{\mathcal{G}'}]^{d'-1-k} (C \ln^4(d) d^{-\ln(d)})^k \leq \left(\frac{d'e}{k}\right)^k (C \ln^4(d) d^{-\ln(d)})^k.$$

For  $k \geq 1$  this is certainly bounded by  $\mathcal{O}(d^{-\ln(d)+1})$ , and we have

$$\begin{aligned} \sum_{k=1}^{d'-1} \binom{d'-1}{k} \Pr[E_{\mathcal{G}'}]^{d'-1-k} (C \ln^4(d) d^{-\ln(d)})^k \\ = \Pr[E_{\mathcal{G}'}]^{d'-1} + \mathcal{O}(d^{-\ln(d)+2}) \end{aligned}$$

We note that the last asymptotic approximation is very rough but sufficient for our purposes. This means that

$$\Pr[h_D(x) = h_D(y)] = 2d' \mathbb{E}_{(\tilde{\mathcal{G}}_x)_1, (\tilde{\mathcal{G}}_y)_1} (\Pr[E_{\mathcal{G}'}]^{d'-1}) + \mathcal{O}(md^{-\ln(d)+2}). \quad (18)$$

Using the same technique as above where we replace the Gaussian density function with  $P_X(x)$ , we have

$$\begin{aligned} \Pr[h'_D(x) = h'_D(y)] &= 2d' \mathbb{E}_{(\mathcal{G}'_x)_1, (\mathcal{G}'_y)_1} (\Pr[E_{\mathcal{G}'}]^{d'-1}) \\ &= 2d' \mathbb{E}_{(\tilde{\mathcal{G}}_x)_1, (\tilde{\mathcal{G}}_y)_2} (\Pr[E_{\mathcal{G}'}] + \mathcal{O}(m2^{-b}))^{d'-1} \\ &= 2d' \mathbb{E}_{(\tilde{\mathcal{G}}_x)_1, (\tilde{\mathcal{G}}_y)_2} (\Pr[E_{\mathcal{G}'}]^{d'-1}) + \mathcal{O}(md^{-\ln(d)+2}) \end{aligned}$$

Finally, plugging this into (18), we get

$$\begin{aligned} \Pr[h_D(x) = h_D(y)] &= \Pr[h'_D(x) = h'_D(y)] + \mathcal{O}(md^{-\ln(d)+2}) \\ &= \Pr[h'_D(x) = h'_D(y)] + \mathcal{O}(d^{-\ln(d)+3}). \end{aligned}$$

Now, we know that by Theorem 3,  $\ln(\Pr[h_D(x) = h_D(y)]) = -\frac{\tilde{R}^2}{4-\tilde{R}^2} \ln(d') + \mathcal{O}_{\tilde{R}}(\ln(\ln d'))$ , so provided  $d$  is large enough that  $\ln(d) - 2 > \frac{\tilde{R}^2}{4-\tilde{R}^2}$ , the lemma follows.