Binary Embedding: Fundamental Limits and Fast Algorithm

Xinyang Yi
The University of Texas at Austin
yixy@utexas.edu

Constantine Caramanis
The University of Texas at Austin
constantine@utexas.edu

Eric Price
The University of Texas at Austin
ecprice@cs.utexas.edu

Abstract

Binary embedding is a nonlinear dimension reduction methodology where high dimensional data are embedded into the Hamming cube while preserving the structure of the original space. Specifically, for an arbitrary N distinct points in $\mathbb{S}^{p-1}$, our goal is to encode each point using m-dimensional binary strings such that we can reconstruct their geodesic distance up to δ uniform distortion. Existing binary embedding algorithms either lack theoretical guarantees or suffer from running time $O\big(mp\big)$. We make three contributions: (1) we establish a lower bound that shows any binary embedding oblivious to the set of points requires $m = \Omega(\frac{1}{\delta^2}\log N)$ bits and a similar lower bound for non-oblivious embeddings into Hamming distance; (2) we propose a novel fast binary embedding algorithm with provably optimal bit complexity $m = O\big(\frac{1}{\delta^2}\log N\big)$ and near linear running time $O(p\log p)$ whenever $\log N \ll \delta\sqrt{p}$, with a slightly worse running time for larger $\log N$; (3) we also provide an analytic result about embedding a general set of points $K \subseteq \mathbb{S}^{p-1}$ with even infinite size. Our theoretical findings are supported through experiments on both synthetic and real data sets.

1 Introduction

Low distortion embeddings that transform high-dimensional points to low-dimensional space have played an important role in dealing with storage, information retrieval and machine learning problems for modern datasets. Perhaps one of the most famous results along these lines is the Johnson-Lindenstrauss (JL) lemma Johnson and Lindenstrauss (1984), which shows that N points can be embedded into a $O\big(\delta^{-2}\log N\big)$-dimensional space while preserving pairwise Euclidean distance up to δ-Lipschitz distortion. This δ^{-2} dependence has been shown to be information-theoretically optimal Alon (2003). Significant work has focused on fast algorithms for computing the embeddings, e.g., (Ailon and Chazelle, 2006; Krahmer and Ward, 2011; Ailon and Liberty, 2013; Cheraghchi et al., 2013; Nelson et al., 2014).

More recently, there has been a growing interest in designing binary codes for high dimensional points with low distortion, i.e., embeddings into the binary cube (Weiss et al., 2009; Raginsky and Lazebnik, 2009; Salakhutdinov and Hinton, 2009; Liu et al., 2011; Gong and Lazebnik, 2011; Yu et al., 2014). Compared to JL embedding, embedding into the binary cube (also called binary embedding) has two advantages in practice: (i) As each data point is represented by a binary code, the disk size for storing the entire dataset is reduced considerably. (ii) Distance in binary cube is some function of the Hamming distance, which can be computed quickly using computationally efficient bit-wise operators. As a consequence, binary embedding can be applied to a large number of domains such as biology, finance and computer vision where the data are usually high dimensional.

While most JL embeddings are linear maps, any binary embedding is fundamentally a nonlinear transformation. As we detail below, this nonlinearity poses significant new technical challenges for both upper and lower bounds. In particular, our understanding of the landscape is significantly less complete. To the best of our knowledge, lower bounds are not known; embedding algorithms for infinite sets have distortion-dependence δ significantly exceeding their finite-set counterparts; and perhaps most significantly, there are no fast (near linear-time) embedding algorithms with strong performance guarantees. As we explain below, this paper contributes to each of these three areas. First, we detail some recent work and state of the art results.

Recent Work. A common approach pursued by several existing works, considers the natural extension of JL embedding techniques via one bit quantization of the projections:

$$\boldsymbol{b}(\boldsymbol{x}) = \text{sign}(\mathbf{A}\boldsymbol{x}), \tag{1.1}$$

where $\boldsymbol{x} \in \mathbb{R}^p$ is input data point, $\mathbf{A} \in \mathbb{R}^{m\times p}$ is a projection matrix and $\boldsymbol{b}(\boldsymbol{x})$ is the embedded binary code. In particular, Jacques et al. (2011) shows when each entry of $\mathbf{A}$ is generated independently from $\mathcal{N}(0,1)$, with $m > \frac{1}{\delta^2}\log N$ it with high probability achieves at most δ (additive) distortion for N points. Work in Plan and Vershynin (2014) extend these results to arbitrary sets $K \subseteq \mathbb{S}^{p-1}$ where $|K|$ can be infinite. They prove that the embedding with δ-distortion can be obtained when $m \gtrsim w(K)^2/\delta^6$ where $w(K)$ is the *Gaussian Mean Width* of K. It is unknown whether the unusual δ^{-6} dependence is optimal or not. Despite provable sample complexity guarantees, one bit quantization of random projection as in (1.1), suffers from $O\big(mp\big)$ running time for a single point. This quadratic dependence can result in a prohibitive computational cost for high-dimensional data. Analogously to the developments in "fast" JL embeddings, there are several algorithms proposed to overcome this computational issue. Work in Gong et al. (2013) proposes a bilinear projection method. By setting $m = O(p)$, their method reduces the running time from $O(p^2)$ to $O(p^{1.5})$. More recently, work in Yu et al. (2014) introduces a circulant random projection algorithm that requires running time $O\big(p\log p\big)$. While these algorithms have reduced running time, as of yet they come without performance guarantees: to the best of our knowledge, the measurement complexities of the two algorithms are still unknown. Another line of work considers learning binary codes from data by solving certain optimization problems (Weiss et al., 2009; Salakhutdinov and Hinton, 2009; Norouzi et al., 2012; Yu et al., 2014). Unfortunately, there is no known provable bits complexity result for these algorithms. It is also worth noting that Raginsky and Lazebnik (2009) provide a binary code design for preserving shift-invariant kernels. Their method suffers from the same quadratic computational issue compared with the fully random Gaussian projection method.

Another related dimension reduction technique is locality sensitive hashing (LSH) where the goal is to compute a discrete data structure such that similar points are mapped into the same bucket with high probability (see, e.g., Andoni and Indyk (2006)). The key difference is that LSH preserves short distances, but binary embedding preserves both short and far distances. For points that are far apart, LSH only cares that the hashings are different while binary embedding cares how different they are.

Contributions of this paper. In this paper, we address several unanswered problems about binary embedding. We provide lower bounds for both data-oblivious and data-aware embeddings; we provide a fast algorithm for binary embedding; and finally we consider the setting of infinite sets, and prove that in some of the most common cases we can improve the state-of-the-art sample complexity guarantees by a factor of δ^{-2}:

1. We provide two lower bounds for binary embeddings. The first shows that any method for embedding and for recovering a distance estimate from the embedded points that is independent of the data being embedded must use $\Omega(\frac{1}{\delta^2}\log N)$ bits. This is based on a bound on the communication complexity of Hamming distance used by Jayram and Woodruff (2013) for a lower bound on the "distributional" JL embedding. Separately, we give a lower bound for arbitrarily data-dependent methods that embed into (any function of) the Hamming distance, showing such algorithms require $m = \Omega(\frac{1}{\delta^2 \log(1/\delta)}\log N)$. This bound is similar to Alon (2003) which gets the same result for JL, but the binary embedding requires a different construction.

2. We provide the first provable fast algorithm with optimal measurement complexity $O\big(\frac{1}{\delta^2}\log N\big)$. The proposed algorithm has running time $O\big(\frac{1}{\delta^2}\log\frac{1}{\delta}\log^2 N\log p\log^3\log N + p\log p\big)$ thus has almost linear time complexity when $\log N \lesssim \delta\sqrt{p}$. Our algorithm is based on two key novel ideas. First, our similarity is based on the median Hamming distance of sub-blocks of the binary code; second, our new embedding takes advantage of a *pair-wise independence argument* of Gaussian Toeplitz projection that could be of independent interest.

3. For arbitrary set $K \subseteq \mathbb{S}^{p-1}$ and the fully random Gaussian projection algorithm, we prove that $m = O(w(K^+)^2/\delta^4)$ is sufficient to achieve δ uniform distortion. Here K^+ is an *expanded* set of K. Although in general $K \subseteq K^+$ and hence $w(K) \le w(K^+)$, for interesting K such as sparse or low rank sets, one can show $w(K^+) = \Theta(w(K)) \ll p$. Therefore applying our theory to these sets results in an improved dependence on δ compared to a recent result in Plan and Vershynin (2014). See Section 3.3 for a detailed discussion.

Discussion. For the fast binary embedding, one simple solution, to the best of our knowledge not previously stated, is to combine a Gaussian projection and the well known results about fast JL. In detail, consider the strategy $\boldsymbol{b}(\boldsymbol{x}) = \mathrm{sign}(\mathbf{AF}\boldsymbol{x})$, where $\mathbf{A}$ is a Gaussian matrix and $\mathbf{F}$ is any fast JL construction such as subsampled Walsh-Hadamard matrix Rudelson and Vershynin (2008) or partial circulant matrix Krahmer et al. (2014) with column flips. A simple analysis shows that this approach achieves measurement complexity $O(\frac{1}{\delta^2}\log N)$ and running time $O(\frac{1}{\delta^4}\log^2 N\log p\log^3\log N + p\log p)$ by following the best known fast JL results. Our fast binary embedding algorithm builds on this simple but effective thought. Instead of using a Gaussian matrix after the fast JL transform, we use a series of Gaussian Toeplitz matrices that have fast matrix

vector multiplication. This novel construction improves the running time by δ^2 while keeping measurement complexity the same. In order for this to work, we need to change the estimator from straight Hamming distance to one based on the median of several Hamming distances.

An interesting point of comparison is Ailon and Rauhut (2014), which considers "RIP-optimal" distributions that give JL embeddings with optimal measurement complexity $O(\frac{1}{\delta^2}\log N)$ and running time $O(p\log p)$. They show the existence of such embeddings whenever $\log N < \delta^2 p^{1/2-\gamma}$ for any constant $\gamma > 0$, which is essentially no better than the bound given by the folklore method of composing a Gaussian projection with a subsampled Fourier matrix. In our binary setting, we show how to improve the region of optimality by a factor of δ. It would be interesting to try and translate this result back to the JL setting.

Notation. We use $[n]$ to denote natural number set $\{1, 2, \ldots, n\}$. For natural numbers $a < b$, let $[a, b]$ denote the consecutive set $\{a, a+1, \ldots, b\}$. A vector in $\mathbb{R}^n$ is denoted as $\boldsymbol{x}$ or equivalently $(x_1, x_2, \ldots, x_n)^\top$. We use $\boldsymbol{x}_\mathcal{I}$ to denote the sub-vector of $\boldsymbol{x}$ with index set $\mathcal{I} \subseteq [n]$. We denote entry-wise vector multiplication as $\boldsymbol{x} \odot \boldsymbol{y} = (x_1y_1, x_2y_2, \ldots, x_ny_n)^\top$. A matrix is typically denoted as $\mathbf{M}$. Term (i, j) of $\mathbf{M}$ is denoted as $\mathbf{M}_{i,j}$. Row i of $\mathbf{M}$ is denoted as $\mathbf{M}_i$. An n-by-n identity matrix is denoted as $\mathbf{I}_n$. For two random variables X, Y, we denote the statement that X and Y are independent as $X \perp Y$. For two binary strings $\boldsymbol{a}, \boldsymbol{b} \in \{0,1\}^m$, we use $d_\mathcal{H}(\boldsymbol{a}, \boldsymbol{b})$ to denote the normalized Hamming distance, i.e., $d_\mathcal{H}(\boldsymbol{a}, \boldsymbol{b}) := \frac{1}{m}\sum_{i=1}^m \mathbb{1}(a_i \neq b_i)$.

2 Organization, Problem Setup and Preliminaries

In this section, we state our problem formally, give some key definitions and present a simple (known) algorithm that sets the stage for the main results of this paper. The algorithm (Algorithm 1), discussed in detail below, is simply the one-bit quantization of a standard JL embedding. Its performance *on finite* sets is easy to analyze, and we state it in Proposition 2.2 below. Three important questions remain unanswered: (i) Lower Bounds – is the performance guaranteed by Proposition 2.2 optimal? We answer this affirmatively in Section 3.1. (ii) Fast Embedding – whereas Algorithm 1 is quadratic (depending on the product mp), fast JL algorithms are nearly linear in p; does something similar exist for binary embedding? We develop a new algorithm in Section 3.2 that addresses the complexity issue, while at the same time guaranteeing δ-embedding with dimension scaling that matches our lower bound. Interestingly, a key aspect of our contribution is that we use a slightly modified similarity function, using the median of the normalized Hamming distance on sub-blocks. (iii) Infinite Sets – recent work analyzing the setting of infinite sets $K \subseteq \mathbb{S}^{p-1}$ shows a dependence of δ^{-6} on the distortion. Is this optimal? We show in Section 3.3 that in many settings this can be improved by a factor of δ^{-2}. In Section 4, we provide numerical results. We give most proofs in Section 5.

2.1 Problem Setup

Given a set of p-dimensional points, our goal is to find a transformation $f : \mathbb{R}^p \mapsto \{0,1\}^m$ such that the Hamming distance (or other related, easily computable metric) between two binary codes is close to their similarity in the original space. We consider points on the unit sphere $\mathbb{S}^{p-1}$ and use

the normalized geodesic distance (occasionally, and somewhat misleadingly, called cosine similarity) as the input space similarity metric. For two points $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^p$, we use $d(\boldsymbol{x}, \boldsymbol{y})$ to denote the geodesic distance, defined as

$$d(\boldsymbol{x}, \boldsymbol{y}) := \frac{\angle(\boldsymbol{x}/\|\boldsymbol{x}\|_2, \boldsymbol{y}/\|\boldsymbol{y}\|_2)}{\pi},$$

where $\angle(\cdot, \cdot)$ denotes the angle between two vectors. For $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{S}^{p-1}$, the metric $d(\boldsymbol{x}, \boldsymbol{y})$ is proportional to the length of the shortest path connecting $\boldsymbol{x}, \boldsymbol{y}$ on the sphere.

Given the success of JL embedding, a natural approach is to consider the one bit quantization of a random projection:

$$\boldsymbol{b} = \operatorname{sign}(\mathbf{A}\boldsymbol{x}), \tag{2.1}$$

where $\mathbf{A}$ is some random projection matrix. Given two points $\boldsymbol{x}, \boldsymbol{y}$ with embedding vectors $\boldsymbol{b}$, and $\boldsymbol{c}$, we have $b_i \neq c_i$ if and only if $\langle \mathbf{A}_i, \boldsymbol{x} \rangle \langle \mathbf{A}_i, \boldsymbol{y} \rangle < 0$. The traditional metric in the embedded space has been the so-called normalized Hamming distance, which we done by $d_{\mathbf{A}}(\boldsymbol{x}, \boldsymbol{y})$ and is defined as follows.

$$d_{\mathbf{A}}(\boldsymbol{x}, \boldsymbol{y}) := \frac{1}{m} \sum_{i=1}^{m} \mathbb{1}\Big\{ \operatorname{sign}\big(\langle \mathbf{A}_i, \boldsymbol{x} \rangle\big) \neq \operatorname{sign}\big(\langle \mathbf{A}_i, \boldsymbol{y} \rangle\big) \Big\}. \tag{2.2}$$

Definition 2.1. (δ-uniform Embedding) Given a set $K \subseteq \mathbb{S}^{p-1}$ and projection matrix $\mathbf{A} \in \mathbb{R}^{m \times p}$, we say the embedding $\boldsymbol{b} = \operatorname{sign}(\mathbf{A}\boldsymbol{x})$ provides a δ-uniform embedding for points in K if

$$\big| d_{\mathbf{A}}(\boldsymbol{x}, \boldsymbol{y}) - d(\boldsymbol{x}, \boldsymbol{y}) \big| \leq \delta, \ \forall \ x, y \in K. \tag{2.3}$$

Note that unlike for JL, we aim to control *additive* error instead of *relative* error. Due to the inherently limited resolution of binary embedding, controlling relative error would force the embedding dimension m to scale inversely with the minimum distance of the original points, and in particular would be impossible for any infinite set.

2.2 Uniform Random Projection

Algorithm 1 Uniform Random Projection

input Finite number of points $K = \{\boldsymbol{x}_i\}_{i=1}^{|K|}$ where $K \subseteq \mathbb{S}^{p-1}$, embedding target dimension m.

1: Construct matrix $\mathbf{A} \in \mathbb{R}^{m \times p}$ where each entry $\mathbf{A}_{i,j}$ is drawn independently from $\mathcal{N}(0, 1)$.
2: **for** $i = 1, 2, \ldots, |K|$ **do**
3: $\quad \boldsymbol{b}_i \leftarrow \operatorname{sign}(\mathbf{A}\boldsymbol{x}_i)$.
4: **end for**

output $\{\boldsymbol{b}_i\}_{i=1}^{|K|}$

Algorithm 1 presents (2.1) formally, when $\mathbf{A}$ is an i.i.d. Gaussian random matrix, i.e., $\mathbf{A}_i \sim \mathcal{N}(0, \mathbf{I}_p)$ for any $i \in [m]$. It is easy to observe that for two fixed points $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{S}^{p-1}$ we have

$$\mathbb{E}\left(\mathbb{1}\left\{ \operatorname{sign}\left(\langle \mathbf{A}_i, \boldsymbol{x} \rangle\right) \neq \operatorname{sign}\left(\langle \mathbf{A}_i, \boldsymbol{y} \rangle\right) \right\} \right) = d(\boldsymbol{x}, \boldsymbol{y}), \ \forall \ i \in [m]. \tag{2.4}$$

The above equality has a geometric explanation: each $\mathbf{A}_i$ actually represents a uniformly distributed random hyperplane in $\mathbb{R}^p$. Then $\text{sign}\left(\langle \mathbf{A}_i, \boldsymbol{x} \rangle\right) \neq \text{sign}\left(\langle \mathbf{A}_i, \boldsymbol{y} \rangle\right)$ holds if and only if hyperplane $\mathbf{A}_i$ intersects the arc between $\boldsymbol{x}$ and $\boldsymbol{y}$. In fact, $d_{\mathbf{A}}(\boldsymbol{x}, \boldsymbol{y})$ is equal to the fraction of such hyperplanes. Under such uniform tessellation, the probability with which the aforementioned event occurs is $d(\boldsymbol{x}, \boldsymbol{y})$. Applying Hoeffding's inequality and probabilistic union bound over N^2 pairs of points, we have the following straightforward guarantee.

Proposition 2.2. Given a set $K \subseteq \mathbb{S}^{p-1}$ with finite size $|K|$, consider Algorithm 1 with $m \geq c(1/\delta^2) \log |K|$. Then with probability at least $1 - 2\exp(-\delta^2 m)$, we have

$$\left| d_{\mathbf{A}}(\boldsymbol{x}, \boldsymbol{y}) - d(\boldsymbol{x}, \boldsymbol{y}) \right| \leq \delta, \ \forall \, \boldsymbol{x}, \boldsymbol{y} \in K.$$

Here c is some absolute constant.

Proof. The proof idea is standard and follows from the above; we omit the details. □

3 Main Results

We now present our main results on lower bounds, on fast binary embedding, and finally, on a general result for infinite sets.

3.1 Lower Bounds

We offer two different lower bounds. The first shows that any embedding technique that is oblivious to the input points must use $\Omega(\frac{1}{\delta^2} \log N)$ bits, regardless of what method is used to estimate geodesic distance from the embeddings. This shows that uniform random projection and our fast binary embedding achieve optimal bit complexity (up to constants). The bound follows from results by Jayram and Woodruff (2013) on the communication complexity of Hamming distance.

Theorem 3.1. Consider any distribution on embedding functions $f : \mathbb{S}^{p-1} \to \{0,1\}^m$ and reconstruction algorithms $g : \{0,1\}^m \times \{0,1\}^m \to \mathbb{R}$ such that for any $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N \in \mathbb{S}^{p-1}$ we have

$$\left| g(f(\boldsymbol{x}_i), f(\boldsymbol{x}_j)) - d(\boldsymbol{x}_i, \boldsymbol{x}_j) \right| \leq \delta$$

for all $i, j \in [N]$ with probability $1 - \epsilon$. Then $m = \Omega(\frac{1}{\delta^2} \log(N/\epsilon))$.

Proof. See Section 5.1 for detailed proof. □

One could imagine, however, that an embedding could use knowledge of the input point set to embed any specific set of points into a lower-dimensional space than is possible with an oblivious algorithm. In the Johnson-Lindenstrauss setting, Alon (2003) showed that this is not possible beyond (possibly) a $\log(1/\delta)$ factor. We show the analogous result for binary embeddings. Relative to Theorem 3.1, our second lower bound works for data-dependent embedding functions but loses a $\log(1/\delta)$ and requires the reconstruction function to depend only on the Hamming distance between the two strings. This restriction is natural because an unrestricted data-dependent reconstruction function could simply encode the answers and avoid any dependence on δ.

With the scheme given in (2.1), choosing $\mathbf{A}$ as a fully random Gaussian matrix yields $d_{\mathbf{A}}(\boldsymbol{x}, \boldsymbol{y}) \approx d(\boldsymbol{x}, \boldsymbol{y})$. However, an arbitrary binary embedding algorithm may not yield a linear functional relationship between Hamming distance and geodesic distance. Thus for this lower bound, we allow the design of an algorithm with arbitrary link function $\mathcal{L}$.

Definition 3.2. (Data-dependent binary embedding problem)
Let $\mathcal{L} : [0, 1] \to [0, 1]$ be a monotonic and continuous function. Given a set of points $\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N \in \mathbb{S}^{p-1}$, we say a binary embedding mapping f solves the binary embedding problem in terms of link function $\mathcal{L}$, if

$$\left| d_{\mathcal{H}}(f(\boldsymbol{x}_i), f(\boldsymbol{x}_j)) - \mathcal{L}\big(d(\boldsymbol{x}_i, \boldsymbol{x}_j)\big) \right| \leq \delta, \ \forall \ i, j \in [N]. \tag{3.1}$$

Although the choice of $\mathcal{L}$ is flexible, note that for the same point, we always have $d_{\mathcal{H}}(f(\boldsymbol{x}_i), f(\boldsymbol{x}_i)) = d(\boldsymbol{x}_i, \boldsymbol{x}_i) = 0$, thus (3.1) implies $\mathcal{L}(0) < \delta$. We can just let $\mathcal{L}(0) = 0$. In particular, we let $\mathcal{L}_{\max} = \mathcal{L}(1)$. We have the following lower bound:

Theorem 3.3. There exist $2N$ points $\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_{2N} \in \mathbb{S}^{N-1}$ such that for any binary embedding algorithm f on $\{\boldsymbol{x}_i\}_{i=1}^{2N}$, if it solves the data-dependent binary embedding problem defined in 3.2 in terms of link function $\mathcal{L}$ and any $\delta \in (0, \frac{1}{16\sqrt{e}}\mathcal{L}_{\max})$, it must satisfy

$$m \geq \frac{1}{128e}\left(\frac{\mathcal{L}_{\max}}{\delta}\right)^2 \frac{\log N}{\log \frac{\mathcal{L}_{\max}}{2\delta}}. \tag{3.2}$$

Proof. See Section 5.2 for detailed proof. □

Remark 3.4. We make two remarks for the above result. (1) When $\mathcal{L}_{\max}$ is some constant, our result implies that for general N points, any binary embedding algorithm (even data-dependent) must have $\Omega(\frac{1}{\delta^2 \log \frac{1}{\delta}} \log N)$ number of measurements. This is analogous to Alon's lower bound in the JL setting. It is worth highlighting two differences: (i) The JL setting considers the same metric (Euclidean distance) for both the input and the embedded spaces. In binary embedding, however, we are interested in showing the relationship between Hamming distance and geodesic distance. (ii) Our lower bound is applicable to a broader class of binary embedding algorithms as it involves arbitrary, even data-dependent, link function $\mathcal{L}$. Such an extension is not considered in the lower bound of JL. (2) The stated lower bound only depends on $\mathcal{L}_{\max}$ and does not depend on any curvature information of $\mathcal{L}$. The constraint $\mathcal{L}_{\max} > 16\sqrt{e}\delta$ is critical for our lower bound to hold, but some such restriction is necessary because for $\mathcal{L}_{\max} < \delta$, we are able to embed all points into just one bit. In this case $d_{\mathcal{H}}(f(\boldsymbol{x}_i), f(\boldsymbol{x}_j)) = 0$ for all pairs and condition (3.1) would hold trivially.

3.2 Fast Binary Embedding

In this section, we present a novel fast binary embedding algorithm. We then establish its theoretical guarantees. There are two key ideas that we leverage: (i) instead of normalized Hamming distance, we use a related metric, the median of the normalized Hamming distance applied to sub-blocks; and (ii) we show a key pair-wise independence lemma for partial Gaussian Toeplitz projection, that allows us to use a concentration bound that then implies nearness in the median-metric we use.

3.2.1 Method

Our algorithm builds on sub-sampled Walsh-Hadamard matrix and partial Gaussian Toeplitz matrices with random column flips. In particular, an m-by-p partial Walsh-Hadamard matrix has the form

$$\mathbf{\Phi} := \mathbf{P} \cdot \mathbf{H} \cdot \mathbf{D}. \tag{3.3}$$

The above construction has three components. We characterize each term as follows:

- Term $\mathbf{D}$ is a p-by-p diagonal matrix with diagonal terms $\{\zeta_i\}_{i=1}^{p}$ that are drawn from i.i.d. Rademacher sequence, i.e, for any $i \in [p]$, $\Pr(\zeta_i = 1) = \Pr(\zeta_i = -1) = 1/2$.
- Term $\mathbf{H}$ is a p-by-p scaled Walsh-Hadamard matrix such that $\mathbf{H}^\top \mathbf{H} = \mathbf{I}_p$.
- Term $\mathbf{P}$ is an m-by-p sparse matrix where one entry of each row is set to be 1 while the rest are 0. The nonzero coordinate of each row is drawn independently from uniform distribution. In fact, the role of $\mathbf{P}$ is to randomly select p rows of $\mathbf{H} \cdot \mathbf{D}$.

An m-by-n partial Gaussian Toeplitz matrix has the form

$$\mathbf{\Psi} := \mathbf{P} \cdot \mathbf{T} \cdot \mathbf{D}. \tag{3.4}$$

We introduce each term as follows:

- Term $\mathbf{D}$ a is n-by-n diagonal matrix with diagonal terms $\{\zeta_i\}_{i=1}^{n}$ that are drawn from i.i.d. Rademacher sequence.
- Term $\mathbf{T}$ is a n-by-n Toeplitz matrix constructed from $(2n-1)$-dimensional vector $\boldsymbol{g}$ such that $\mathbf{T}_{i,j} = g_{i-j+n}$ for any $i, j \in [n]$. In particular, $\boldsymbol{g}$ is drawn from $\mathcal{N}(0, \mathbf{I}_{2n-1})$.
- Term $\mathbf{P}$ is an m-by-n sparse matrix where $\mathbf{P}_i = \boldsymbol{e}_i^\top$ for any $i \in [m]$. Equivalently, we use $\mathbf{P}$ to select the first m rows of $\mathbf{TD}$. It's worth to note we actually only need to select any distinct m rows.

With the above constructions in hand, we present our fast algorithm in Algorithm 2. At a high level, Algorithm 2 consists of two parts: First, we apply column flipped partial Hadamard transform to convert p-dimensional point into n-dimensional intermediate point. Second, we use B independent (m/B)-by-n partial Gaussian Toeplitz matrices and sign operator to map an intermediate point into B blocks of binary codes. In terms of similarity computation for the embedded codes, we use the median of each block's normalized Hamming distance. In detail, for $\boldsymbol{b}, \boldsymbol{c} \in \{0,1\}^m$, B-wise normalized Hamming distance is defined as

$$d_{\mathcal{H}}(\boldsymbol{b}, \boldsymbol{c}; B) := \text{median}\left(\left\{d_{\mathcal{H}}\left(\boldsymbol{b}_{T_i}, \boldsymbol{c}_{T_i}\right)\right\}_{i=0}^{B-1}\right) \tag{3.5}$$

where $T_i = [i+1, i+m/B]$.

It is worth noting that our first step is one construction of fast JL transform. In fact any fast JL transform would work for our construction, but we choose a standard one with real value: based on

Rudelson and Vershynin (2008); Cheraghchi et al. (2013); Krahmer and Ward (2011), it is known that with $m = O\big(\epsilon^{-2} \log N \log p \log^3(\log N)\big)$ measurements, a subsampled Hadamard matrix with column flips becomes an ϵ-JL matrix for N points.

The second part of our algorithm follows framework (2.1). By choosing a Gaussian random vector in each row of Ψ, from our previous discussion in Section 2.2, the probability that such a hyperplane intersects the arc between two points is equal to their geodesic distance. Compared to a fully random Gaussian matrix, as used in Algorithm 1, the key difference is that the hyperplanes represented by rows of Ψ are not independent to each other; this imposes the main analytical challenge.

Algorithm 2 Fast Binary Embedding

input Finite number of points $\{\boldsymbol{x}_i\}_{i=1}^N$ where each point $\boldsymbol{x}_i \in \mathbb{S}^{p-1}$, embedded dimension m, intermediate dimension n, number of blocks B.

Draw a n-by-p sub-sampled Walsh-Hadamard matrix $\boldsymbol{\Phi}$ according to (3.3). Draw B independent partial Gaussian Toeplitz matrices $\{\boldsymbol{\Psi}^{(j)}\}_{j=1}^B$ with size (m/B)-by-n according to (3.4).
$\{$*Part I: Fast JL*$\}$
for $i = 1, 2, \ldots, N$ **do**
  $\boldsymbol{y}_i \leftarrow \boldsymbol{\Phi} \cdot \boldsymbol{x}_i$.
end for
$\{$*Part II: Partial Gaussian Toeplitz Projection*$\}$
for $i = 1, 2, \ldots, N$ **do**
  for $j = 1, 2, \ldots, B$ **do**
    $\boldsymbol{c}_j \leftarrow \operatorname{sign}\left(\boldsymbol{\Psi}^{(j)} \cdot \boldsymbol{y}_i\right)$.
  end for
  $\boldsymbol{b}_i \leftarrow [\boldsymbol{c}_1; \boldsymbol{c}_2; \ldots; \boldsymbol{c}_B]$
end for

output $\{\boldsymbol{b}_i\}_{i=1}^N$

3.2.2 Analysis

We give the analysis for Algorithm 2. We first review a well known result about fast JL transform.

Lemma 3.5. Consider the column flipped partial Hadamard matrix defined in (3.3) with size m-by-p. For N points $\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N \in \mathbb{S}^{p-1}$, let $\boldsymbol{y}_i = \sqrt{\frac{p}{m}}\Phi(\boldsymbol{\zeta}) \cdot \boldsymbol{x}_i, \ \forall\ i \in [N]$. For some absolute constant c, suppose $m \geq c\delta^{-2} \log N \log p \log^3(\log N)$, then with probability at least 0.99, we have that for any $i, j \in [N]$

$$\big|\|\boldsymbol{y}_i - \boldsymbol{y}_j\|_2 - \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2\big| \leq \delta \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2, \tag{3.6}$$

and for any $i \in [N]$

$$\big|\|\boldsymbol{y}_i\|_2 - 1\big| \leq \delta. \tag{3.7}$$

Proof. It can be proved by combining Theorem 14 in Cheraghchi et al. (2013) and Theorem 3.1 in Krahmer and Ward (2011). □

The above result suggests that the first part of our algorithm reduces the dimension while preserving well the Euclidean distance of each pair. Under this condition, all the pairwise geodesic distances are also well preserved as confirmed by the following result.

Lemma 3.6. Consider the set of embedded points $\{\boldsymbol{y}_i\}_{i=1}^N$ defined in Lemma 3.5. Suppose conditions (3.6)-(3.7) hold with $\delta > 0$. Then for any $i, j \in [N]$,

$$\left|d(\boldsymbol{y}_i, \boldsymbol{y}_j) - d(\boldsymbol{x}_i, \boldsymbol{x}_j)\right| \leq C\delta \tag{3.8}$$

holds with some absolute constant C.

Proof. We postpone the proof to Appendix A. □

The next result is our independence lemma, and is one of the key technical ideas that make our result possible. The result shows that for any fixed $\boldsymbol{x}$, Gaussian Toeplitz projection (with column flips) plus sign$(\cdot)$ generate pair-wise independent binary codes.

Lemma 3.7. Let $\boldsymbol{g} \sim \mathcal{N}(0, \mathbf{I}_{2n-1})$, $\boldsymbol{\zeta} = \{\zeta_i\}_{i=1}^{i=n}$ be an i.i.d. Rademacher sequence. Let $\mathbf{T}$ be a random Toeplitz matrix constructed from $\boldsymbol{g}$ such that $\mathbf{T}_{i,j} = g_{i-j+n}$. Consider any two distinct rows of $\mathbf{T}$ say $\boldsymbol{\xi}$, $\boldsymbol{\xi}'$. For any two fixed vectors $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$, we define the following random variables

$$\begin{aligned} X &= \operatorname{sign}\langle \boldsymbol{\xi} \odot \boldsymbol{\zeta},\ \boldsymbol{x}\rangle,\ \ X' = \operatorname{sign}\langle \boldsymbol{\xi}' \odot\ \boldsymbol{\zeta}, \boldsymbol{x}\rangle; \\ Y &= \operatorname{sign}\langle \boldsymbol{\xi} \odot \boldsymbol{\zeta},\ \boldsymbol{y}\rangle,\ \ \ Y' = \operatorname{sign}\langle \boldsymbol{\xi}' \odot\ \boldsymbol{\zeta}, \boldsymbol{y}\rangle. \end{aligned}$$

We have

$$X \perp X', X \perp Y', Y \perp X', Y \perp Y'.$$

Proof. See Section 5.3.1 for detailed proof. □

We are ready to prove the following result about Algorithm 2.

Theorem 3.8. Consider Algorithm 2 with random matrices $\boldsymbol{\Phi}$, $\boldsymbol{\Psi}$ defined in (3.3) and (3.4) respectively. For finite number of points $\{\boldsymbol{x}_i\}_{i=1}^N$, let $\boldsymbol{b}_i$ be the binary codes of $\boldsymbol{x}_i$ generated by Algorithm 2. Suppose we set

$$B \geq c \log N,\ n \geq c'(1/\delta^2) \log N \log p \log^3(\log N),\ n \geq m/B \geq c''(1/\delta^2),$$

with some absolute constants c, c', c'', then with probability at least 0.98, we have that for any $i, j \in [N]$

$$\left|d_{\mathcal{H}}(\boldsymbol{b}_i, \boldsymbol{b}_j; B) - d(\boldsymbol{x}_i, \boldsymbol{x}_j)\right| \leq \delta.$$

Similarity metric $d_{\mathcal{H}}(\cdot, \cdot; B)$ is the median of normalized Hamming distance defined in (3.5).

Proof. See Section 5.3.2 for detailed proof. □

The above result suggests that the measurement complexity of our fast algorithm is $O\big(\frac{1}{\delta^2}\log N\big)$ which matches the performance of Algorithm 1 based on fully random matrix. Note that this measurement complexity can not be improved significantly by any data-oblivious binary embedding with any similarity metric, as suggested by Theorem 3.1.

Running time: The first part of our algorithm takes time $O\big(p\log p\big)$. Generating a single block of binary codes from partial Toeplitz matrix takes time $O\big(n\log(\frac{1}{\delta})\big)$[1]. Thus the total running time is $O\big(Bn\log\frac{1}{\delta}+p\log p\big)=O\big(\frac{1}{\delta^2}\log\frac{1}{\delta}\log^2 N\log p\log^3(\log N)+p\log p\big)$. By ignoring the polynomial $\log\log$ factor, the second term $O\big(p\log p\big)$ dominates when $\log N\lesssim\delta\sqrt{p/\log\frac{1}{\delta}}$.

Comparison to an alternative algorithm: Instead of utilizing the partial Gaussian Toeplitz projection, an alternative method, to the best of our knowledge not previously stated, is to use fully random Gaussian projection in the second part of our algorithm. We present the details in Algorithm 3. By combining Proposition 2.2 and Lemma 3.5, it is straightforward to show this algorithm still achieves the same measurement complexity $O\big(\frac{1}{\delta^2}\log N\big)$. The corresponding running time is $O\big(\frac{1}{\delta^4}\log^2 N\log p\log^3(\log N)+p\log p\big)$, so it is fast when $\log N\lesssim\delta^2\sqrt{p}$. Therefore our algorithm has an improved dependence on δ. This improvement comes from fast multiplication of partial Toeplitz matrix and a pair-wise independence argument shown in Lemma 3.7.

Algorithm 3 Alternative Fast Binary Embedding

input Finite number of points $\{\boldsymbol{x}_i\}_{i=1}^N$ where each point $\boldsymbol{x}_i\in\mathbb{S}^{p-1}$, embedded dimension m, intermediate dimension n.

1: Draw a n-by-p sub-sampled Walsh-Hadamard matrix $\boldsymbol{\Phi}$ according to (3.3). Construct m-by-n matrix $\mathbf{A}$ where each entry is drawn independently from $\mathcal{N}(0,1)$.
2: **for** $i=1,2,\ldots,N$ **do**
3: $\quad\boldsymbol{b}_i\leftarrow\mathrm{sign}(\mathbf{A}\boldsymbol{\Phi}\boldsymbol{x}_i)$
4: **end for**

output $\{\boldsymbol{b}_i\}_{i=1}^N$

3.3 δ-uniform Embedding for General K

In this section, we turn back to the fully random projection binary embedding (Algorithm 1). Recall that in Proposition 2.2, we show for finite size K, $m=O(\frac{1}{\delta^2}\log|K|)$ measurements are sufficient to achieve δ-uniform embedding. For general K, the challenge is that there might be an infinite number of distinct points in K, so Proposition 2.2 cannot be applied. In proving the JL lemma for an infinite set K, the standard technique is either constructing an ϵ-net of K or reducing the distortion to the deviation bound of a Gaussian process. However, due to the non-linearity essential for binary embedding, these techniques cannot be directly extended to our setting. Therefore strengthening Proposition 2.2 to infinite size K imposes significant technical challenges. Before stating our result, we first give some definitions.

Definition 3.9. (Gaussian mean width) Let $\boldsymbol{g}\sim\mathcal{N}(0,\mathbf{I}_p)$. For any set $K\subseteq\mathbb{S}^{p-1}$, the Gaussian

[1]Matrix-vector multiplication for m-by-n partial Toeplitz matrix can be implemented in running time $O\big(n\log m\big)$.

mean width of K is defined as

$$w(K) := \mathbb{E}_{\boldsymbol{g}} \sup_{\boldsymbol{x} \in K} \big|\langle \boldsymbol{g}, \boldsymbol{x} \rangle\big|.$$

Here, $w(K)^2$ measures the effective dimension of set K. In the trivial case $K = \mathbb{S}^{p-1}$, we have $w(K)^2 \lesssim p$. However, when K has some special structure, we may have $w(K)^2 \ll p$. For instance, when $K = \{\boldsymbol{x} \in \mathbb{S}^{p-1} : |\operatorname{supp}(\boldsymbol{x})| \le s\}$, it has been shown that $w(K) = \Theta(\sqrt{s \log(p/s)})$ (see Lemma 2.3 in Plan and Vershynin (2013)).

For a given δ, we define K_δ^+, the *expanded version* of $K \subseteq \mathbb{S}^{p-1}$ as:

$$K_\delta^+ := K \bigcup \big\{ \boldsymbol{z} \in \mathbb{S}^{p-1} : \boldsymbol{z} = \frac{\boldsymbol{x} - \boldsymbol{y}}{\|\boldsymbol{x} - \boldsymbol{y}\|_2},\ \forall\, \boldsymbol{x}, \boldsymbol{y} \in K \text{ if } \delta^2 \le \|\boldsymbol{x} - \boldsymbol{y}\|_2 \le \delta \big\}. \tag{3.9}$$

In other words, K_δ^+ is constructed from K by adding the normalized differences between pairs of points in K that are within δ but not closer than δ^2. Now we state the main result as follows.

Theorem 3.10. Consider any $K \subseteq \mathbb{S}^{p-1}$. Let $\mathbf{A} \in \mathbb{R}^{m \times p}$ be an i.i.d. Gaussian matrix where each row $\mathbf{A}_i \sim \mathcal{N}(0, \mathbf{I}_p)$. For any two points $\boldsymbol{x}, \boldsymbol{y} \in K$, $\boldsymbol{d}_{\mathbf{A}}(\boldsymbol{x}, \boldsymbol{y})$ is defined in (2.2). Expanded set K_δ^+ is defined in (3.9). When

$$m \ge c \frac{w(K_\delta^+)^2}{\delta^4},$$

with some absolute constant c, then we have that

$$\sup_{\boldsymbol{x}, \boldsymbol{y} \in K} \big| d_{\mathbf{A}}(\boldsymbol{x}, \boldsymbol{y}) - d(\boldsymbol{x}, \boldsymbol{y}) \big| \le \delta$$

holds with probability at least $1 - c_1 \exp(-c_2 \delta^2 m)$ where c_1, c_2 are absolute constants.

Proof. See Section 5.4 for detailed proof. □

Remark 3.11. We compare the above result to Theorem 1.5 from the recent paper Plan and Vershynin (2014) where it is proved that for $m \gtrsim w(K)^2/\delta^6$, Algorithm 1 is guaranteed to achieve δ-uniform embedding for general K. Based on definition (3.9), we have

$$w(K) \le w(K_\delta^+) \le \frac{1}{\delta^2} w(K - K) \lesssim \frac{1}{\delta^2} w(K).$$

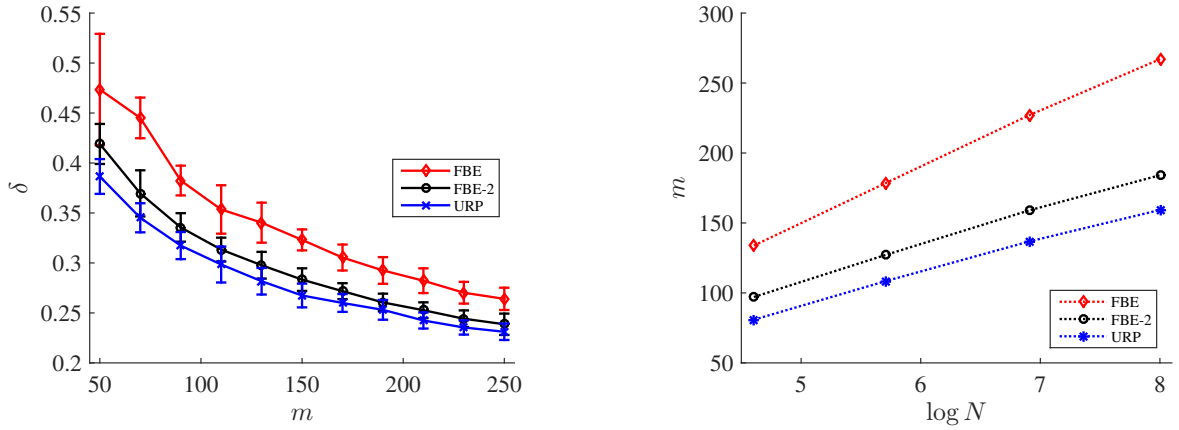
Thus in the worst case, Theorem 3.10 recovers the previous result up to a factor $\frac{1}{\delta^2}$. More importantly, for many interesting sets one can show $w(K_\delta^+) \lesssim w(K)$; in such cases, our result leads to an improved dependence on δ. We give several such examples as follows:

- **Low rank set.** For some $\mathbf{U} \in \mathbb{R}^{p \times r}$ such that $\mathbf{U}^\top \mathbf{U} = \mathbf{I}_r$, let $K = \{\boldsymbol{x} \in \mathbb{S}^{p-1} : \boldsymbol{x} = \mathbf{U}\boldsymbol{c},\ \forall\, \boldsymbol{c} \in \mathbb{S}^{r-1}\}$. We simply have $K = K_\delta^+$ and $w(K) \lesssim \sqrt{r}$. Our result implies $m = O\big(r/\delta^4\big)$.
- **Sparse set.** $K = \{\boldsymbol{x} \in \mathbb{S}^{p-1} : |\operatorname{supp}(\boldsymbol{x})| \le s\}$. In this case we have $K_\delta^+ \subseteq \{\boldsymbol{x} \in \mathbb{S}^{p-1} : |\operatorname{supp}(\boldsymbol{x})| \le 2s\}$. Therefore $w(K_\delta^+) = \Theta(\sqrt{s \log(p/s)})$. Our result implies $m = O\big(\frac{s \log(p/s)}{\delta^4}\big)$.
- **Set with finite size.** $|K| < \infty$. As $w(K) \lesssim \sqrt{\log |K|}$ and $|K_\delta^+| \le 2|K|$, our result implies $m = O\big(\log |K|/\delta^4\big)$. We thus recover Proposition 2.2 up to factor $1/\delta^2$.

Applying the result from Plan and Vershynin (2014) to the above sets implies similar results but the dependence on δ becomes $1/\delta^6$.

4 Numerical Results

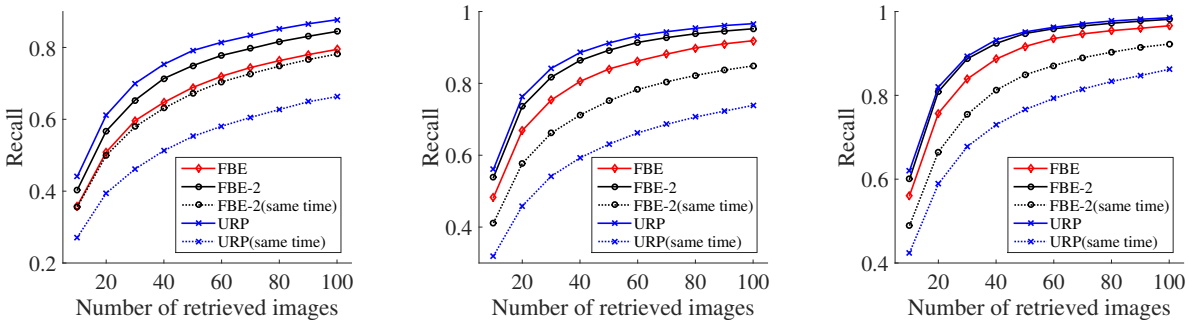
In this section, we present the results of experiments we conduct to validate our theory and compare the performance of the following three algorithms we discussed: uniform random projection (URP) (Algorithm 1), fast binary embedding (FBE) (Algorithm 2) and the alternative fast binary embedding (FBE-2) (Algorithm 3). We first apply these algorithms to synthetic datasets. In detail, given parameters (N, p), a synthetic dataset is constructed by sampling N points from $\mathbb{S}^{p-1}$ uniformly at random. Recall that δ is the maximum embedding distortion among all pairs of points. We use m to denote the number of binary measurements. Algorithm FBE needs parameters n, B, which are intermediate dimension and number of blocks respectively. Based on Theorem 3.8, n is required to be proportional to m (up to some logarithmic factors) and B is required to be proportional to $\log N$. We thus set $n \approx 1.3m$, $B \approx 1.8 \log N$. We also set $n \approx 1.3m$ for FBE-2. In addition, we fix $p = 512$. We report our first result showing the functional relationship between (m, N, δ) in Figure 1. In particular, panel 1(a) shows the the change of distortion δ over the number of measurements m for fixed N. We observe that, for all the three algorithms, δ decays with m at the rate predicted by Proposition 2.2 and Theorem 3.8. Panel 1(b) shows the empirical relationship between m and $\log N$ for fixed δ. As predicted by our theory (lower bound and upper bound), m has a linear dependence on $\log N$.



(a) $N = 300$ (b) $\delta = 0.3$

Figure 1: Results on synthetic datasets. **(a)** Each point, along with the standard deviation represented by the error bar, is an average of 50 trials each of which is based on a fresh synthetic dataset with size $N = 300$ and newly constructed embedding mapping. **(b)** Each point is computed by slicing at $\delta = 0.3$ in similar plots like (a) but with the corresponding N.

A popular application of binary embedding is image retrieval, as considered in (Gong and Lazebnik, 2011; Gong et al., 2013; Yu et al., 2014). We thus conduct an experiment on the Flickr-25600 dataset that consists of $10k$ images from Internet. Each image is represented by a 25600-dimensional normalized Fisher vector. We take 500 randomly sampled images as query points and leave the rest as base for retrieval. The *relevant images* of each query are defined as its 10 nearest neighbors



(a) $m = 5000$ (b) $m = 10000$ (c) $m = 15000$

Figure 2: Image retrieval results on Flickr-25600. Each panel presents the recall for specified number of measurements m. Black and blue dot lines are respectively the recall of FBE-2 and URP with less number of measurements but the same running time as FBE.

based on geodesic distance. Given m, we apply FBE, FBE-2 and URP to convert all images into m-dimensional binary codes. In particular, we set $B = 10$ for FBE and $n \approx 1.3m$ for FBE and FBE-2. Then we leverage the corresponding similarity metrics, (3.5) for FBE and Hamming distance for FBE-2 and URP, to retrieve the nearest images for each query. The performance of each algorithm is characterized by *recall*, i.e., the number of retrieved *relevant* images divided by the total number of relevant images. We report our second result in Figure 2. Each panel shows the average recall of all queries for a specified m. We note that FBE-2, as a fast algorithm, performs as well as URP with the same number of measurements. In order to show the running time advantage of our fast algorithm FBE, we also present the performance of FBE-2 and URP with fewer measurements such that they can be computed with the same time as FBE. As we observe, with large number of measurements, FBE-2 and URP perform marginally better than FBE while FBE has a significant improvement over the two algorithms under identical time constraint.

5 Proofs

5.1 Proof of Data-Oblivious Lower Bound (Theorem 3.1)

The proof of the data-oblivious lower bound is based on a lower bound for one-way communication of Hamming distance due to Jayram and Woodruff (2013).

Definition 5.1 (One-way communication of Hamming distance)**.** In the one-way communication model, Alice is given $\boldsymbol{a} \in \{0,1\}^n$ and Bob is given $\boldsymbol{b} \in \{0,1\}^n$. Alice sends Bob a message $\boldsymbol{c} \in \{0,1\}^m$, and Bob uses $\boldsymbol{b}$ and $\boldsymbol{c}$ to output a value $x \in \mathbb{R}$. Alice and Bob have shared randomness.

Alice and Bob solve the (δ, ϵ) additive Hamming distance estimation problem if $|x - d_{\mathcal{H}}(\boldsymbol{a}, \boldsymbol{b})| \leq \delta$ with probability $1 - \epsilon$.

The result proven in Jayram and Woodruff (2013) is a lower bound for the *multiplicative* Hamming distance estimation problem, but their techniques readily yield a bound for the additive case

as well:

Lemma 5.2. Any algorithm that solves the (δ, ϵ) additive Hamming distance estimation problem must have $m = \Omega((1/\delta^2)\log(1/\epsilon))$ as long as this is less than n.

Proof. We apply Lemma 3.1 of Jayram and Woodruff (2013) with parameters $\alpha = 2$, $p = 1$, $b = 1$, $\varepsilon = \delta$, and $\delta = \epsilon$. This encodes inputs from a problem they prove is hard (augmented indexing on large domains) to inputs appropriate for Hamming estimation. In particular, for $n' = O(\frac{1}{\delta^2}\log(1/\epsilon))$ it gives a distribution on $(\boldsymbol{a}, \boldsymbol{b}) \in \{0,1\}^{n'} \times \{0,1\}^{n'}$ that are divided into "NO" and "YES" instances, such that:

- From the reduction, distinguishing NO instances from YES instances with probability $1 - \epsilon$ requires Alice to send $m = \Omega(\frac{1}{\delta^2}\log(1/\epsilon))$ bits of communication to Bob.
- In NO instances, $d_{\mathcal{H}}(\boldsymbol{a}, \boldsymbol{b}) \geq \frac{1}{2}(1 - \delta/3)$.
- In YES instances, $d_{\mathcal{H}}(\boldsymbol{a}, \boldsymbol{b}) \leq \frac{1}{2}(1 - 2\delta/3)$.

First, suppose $n = n'$. Then since solving the additive Hamming distance estimation problem with $\delta/12$ accuracy would distinguish NO instances from YES instances, it must involve $m = \Omega(\frac{1}{\delta^2}\log(1/\epsilon))$ bits of communication.

For $n > n'$, simply duplicate the coordinates of a and b $\lfloor n/n' \rfloor$ times, and zero-pad the remainder. Less than half the coordinates are then part of the zero-padding, so the gap between YES and NO instances remains at least $\delta/12$ and a protocol for the $(\delta/24, \epsilon)$ additive Hamming distance estimation problem requires $m = \Omega(\frac{1}{\delta^2}\log(1/\epsilon))$ as desired. □

With this in hand, we can prove Theorem 3.1:

Proof of Theorem 3.1. We reduce one-way communication of the (δ, ϵ) additive Hamming distance estimation problem to the embedding problem. Let $\boldsymbol{a}, \boldsymbol{b} \in \{0,1\}^p$ be drawn from the hard instance for the communication problem defined in Lemma 5.2. Linearly transform them to $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{S}^{p-1}$ via $\boldsymbol{u} = (2 \cdot \boldsymbol{a} - \boldsymbol{1})/\sqrt{p}$, $\boldsymbol{v} = (2 \cdot \boldsymbol{b} - \boldsymbol{1})/\sqrt{p}$. We have that $\langle \boldsymbol{u}, \boldsymbol{v} \rangle = 1 - 2d_{\mathcal{H}}(\boldsymbol{a}, \boldsymbol{b})$, so

$$d(\boldsymbol{u}, \boldsymbol{v}) = 1 - \frac{\arccos(\langle \boldsymbol{u}, \boldsymbol{v} \rangle)}{\pi} = 1 - \frac{\arccos(1 - 2d_{\mathcal{H}}(\boldsymbol{a}, \boldsymbol{b}))}{\pi}$$

or

$$d_{\mathcal{H}}(\boldsymbol{a}, \boldsymbol{b}) = \frac{1}{2}(1 - \cos(\pi - \pi d(\boldsymbol{u}, \boldsymbol{v})))$$

Given an estimate of $d(\boldsymbol{u}, \boldsymbol{v})$, we can therefore get an estimate of $d_{\mathcal{H}}(\boldsymbol{a}, \boldsymbol{b})$. In particular, since $|\cos'(x)| \leq 1$, if we learn $d(\boldsymbol{u}, \boldsymbol{v})$ to $\pm\delta$ then we learn $d_{\mathcal{H}}(\boldsymbol{a}, \boldsymbol{b})$ to $\pm\delta\frac{\pi}{2}$.

For now, consider the case of $N = 2$. Consider an oblivious embedding function $f : \mathbb{S}^{p-1} \to \{0,1\}^m$ and reconstruction algorithm $g : \{0,1\}^m \times \{0,1\}^m \to \mathbb{R}$ that has

$$|g(f(\boldsymbol{u}), f(\boldsymbol{v})) - d(\boldsymbol{u}, \boldsymbol{v})| \leq \delta\frac{2}{\pi}$$

with probability $1 - \epsilon$ on the distribution of inputs $(\boldsymbol{u}, \boldsymbol{v})$. We can solve the one-way communication problem for Hamming distance estimation by Alice sending $f(\boldsymbol{u})$ to Bob, Bob learning $d(\boldsymbol{u}, \boldsymbol{v}) \approx$

$g(f(\boldsymbol{u}), f(\boldsymbol{v}))$, and then computing $d_{\mathcal{H}}(\boldsymbol{a}, \boldsymbol{b})$ to $\pm\delta$. By the lower bound for this problem, any such f and g must have $m = \Omega(\frac{1}{\delta^2} \log \frac{1}{\epsilon})$, proving the result for $N = 2$ (after rescaling δ).

For general N, we draw instances $(\boldsymbol{u}_1, \boldsymbol{v}_1), (\boldsymbol{u}_2, \boldsymbol{v}_2), \ldots, (\boldsymbol{u}_{N/2}, \boldsymbol{v}_{N/2})$ independently from the hard instance for binary embedding of $N = 2$ and $\epsilon' = 4\epsilon/N$. Consider an oblivious embedding function $f : \mathbb{S}^{p-1} \to \{0,1\}^m$ and reconstruction algorithm $g : \{0,1\}^m \times \{0,1\}^m \to \mathbb{R}$ that has for all $i \in [N/2]$ that

$$|g(f(\boldsymbol{u}_i), f(\boldsymbol{v}_i)) - d(\boldsymbol{u}_i, \boldsymbol{v}_i)| \leq \delta$$

with probability $1-\epsilon$ on this distribution. Define α to be the probability that $|g(f(\boldsymbol{u}_i), f(\boldsymbol{v}_i)) - d(\boldsymbol{u}_i, \boldsymbol{v}_i)| \leq \delta$ for any particular i. Because f and g are oblivious and the different instances are independent, we have the probability that all instances succeed is $\alpha^{N/2} \geq 1 - \epsilon$, so

$$\alpha > (1-\epsilon)^{2/N} > 1 - 4\epsilon/N.$$

In particular, this means f and g solve the hard instance of binary embedding and $N = 2$, $\epsilon' = 4\epsilon/N$. By the above lower bound for $N = 2$, this means

$$m = \Omega(\frac{1}{\delta^2} \log(N/\epsilon))$$

as desired. □

5.2 Proof of Data-Dependent Lower Bound (Theorem 3.3)

We need a few ingredients to show the lower bound. First, we define a matrix that is close to identity matrix.

Definition 5.3. ((δ_1, δ_2)-near identity matrix) Symmetric matrix $\mathbf{M} \in \mathbb{R}^{p \times p}$ is called a (δ_1, δ_2)-near identity matrix if it satisfies both of the following conditions:

$$1 - \delta_1 \leq \mathbf{M}_{i,i} \leq 1, \forall\, i \in [p],$$

$$\left|\mathbf{M}_{i,j}\right| \leq \delta_2, \forall\, i \neq j \in [p].$$

Next we give a lower bound on the rank of (δ_1, δ_2)-near identity matrix.

Lemma 5.4. Suppose positive semidefinite matrix $\mathbf{M} \in \mathbb{R}^{p \times p}$ is a (δ_1, δ_2)-near identity matrix with rank d, and $0 < \delta_1, \delta_2 < 1$. Then we have

$$d \geq \frac{p(1-\delta_1)^2}{1 + (p-1)\delta_2^2}.$$

Proof. We postpone the proof to Appendix B. □

The above result is weak when it is applied to show our desired lower bound. We still need to make use of the following combinatorial result.

Lemma 5.5. Suppose matrix $\mathbf{M} \in \mathbb{R}^{p\times p}$ has rank d. Let $P(x)$ be any degree k polynomial function. Consider matrix $\mathbf{N} \in \mathbb{R}^{p\times p}$ defined as $\mathbf{N} := P(\mathbf{M})$, where the $\mathbf{N}_{i,j} = P(\mathbf{M}_{i,j})$. We have

$$\operatorname{rank}(\mathbf{N}) \le \binom{k+d}{k}.$$

Proof. See Lemma 9.2 of Alon (2003) for a detailed proof. □

Now we are ready to prove Theorem 3.3.

Proof of Theorem 3.3. Let $\boldsymbol{e}_i$ denote the i'th natural basis of $\mathbb{R}^N$, i.e., the i'th coordinate is 1 while the rest are all zeros. Consider N points $\{\boldsymbol{e}_1, \boldsymbol{e}_2, ..., \boldsymbol{e}_N\}$ and their opposite vectors $\{-\boldsymbol{e}_1, -\boldsymbol{e}_2, ..., -\boldsymbol{e}_N\}$. For any binary embedding algorithm f, we let

$$\boldsymbol{b}_i := f(\boldsymbol{e}_i),\ \forall\ i \in [N],$$

$$\boldsymbol{c}_i := f(-\boldsymbol{e}_i),\ \forall\ i \in [N].$$

Under the condition that f solves the general binary embedding problem with link function $\mathcal{L}$, we have

$$\left|d_{\mathcal{H}}(\boldsymbol{b}_i, \boldsymbol{c}_i) - \mathcal{L}\big(d(\boldsymbol{e}_i, -\boldsymbol{e}_i)\big)\right| \le \delta, \forall\ i \in [N]. \tag{5.1}$$

As $d(\boldsymbol{e}_i, -\boldsymbol{e}_i) = 1$, we have

$$\mathcal{L}(1) + \delta \ge d_{\mathcal{H}}(\boldsymbol{b}_i, \boldsymbol{c}_i) \ge \mathcal{L}(1) - \delta. \tag{5.2}$$

Similarly, note that

$$d(\boldsymbol{e}_i, \boldsymbol{e}_j) = d(\boldsymbol{e}_i, -\boldsymbol{e}_j) = d(-\boldsymbol{e}_i, -\boldsymbol{e}_j) = \frac{1}{2},\ \forall\ i \ne j,$$

we have $\forall\ i \ne j$

$$\mathcal{L}(1/2) - \delta \le d_{\mathcal{H}}(\boldsymbol{b}_i, \boldsymbol{b}_j) \le \mathcal{L}(1/2) + \delta, \tag{5.3}$$

$$\mathcal{L}(1/2) - \delta \le d_{\mathcal{H}}(\boldsymbol{c}_i, \boldsymbol{c}_j) \le \mathcal{L}(1/2) + \delta, \tag{5.4}$$

$$\mathcal{L}(1/2) - \delta \le d_{\mathcal{H}}(\boldsymbol{b}_i, \boldsymbol{c}_j) \le \mathcal{L}(1/2) + \delta. \tag{5.5}$$

From now on, we treat binary strings $\boldsymbol{b}_i, \boldsymbol{c}_i$ as vectors in $\mathbb{R}^m$. Let $\mathbf{B}$ denote the matrix with rows $\boldsymbol{b}_i$ and $\mathbf{C}$ denote the matrix with rows $\boldsymbol{c}_i$. Consider the outer product of the difference between $\mathbf{B}$ and $\mathbf{C}$, namely

$$\mathbf{M} = (\mathbf{B} - \mathbf{C})(\mathbf{B} - \mathbf{C})^\top.$$

Note that $\forall\ i \in [N]$,

$$\mathbf{M}_{i,i} = \|\boldsymbol{b}_i - \boldsymbol{c}_i\|_2^2 = 4m \cdot d_{\mathcal{H}}(\boldsymbol{b}_i, \boldsymbol{c}_i) \ge 4m\big(\mathcal{L}(1) - \delta\big).$$

The last inequality follows from (5.2). For $\forall\ i \ne j$, we have

$$\begin{aligned}\mathbf{M}_{i,j} &= \big\langle \boldsymbol{b}_i - \boldsymbol{c}_i, \boldsymbol{b}_j - \boldsymbol{c}_j \big\rangle = \big\langle \boldsymbol{b}_i, \boldsymbol{b}_j \big\rangle + \big\langle \boldsymbol{c}_i, \boldsymbol{c}_j \big\rangle - \big\langle \boldsymbol{b}_i, \boldsymbol{c}_j \big\rangle - \big\langle \boldsymbol{b}_j, \boldsymbol{c}_i \big\rangle \\ &= 2m\Big(d_{\mathcal{H}}(\boldsymbol{b}_i, \boldsymbol{c}_j) + d_{\mathcal{H}}(\boldsymbol{b}_j, \boldsymbol{c}_i) - d_{\mathcal{H}}(\boldsymbol{b}_i, \boldsymbol{b}_j) - d_{\mathcal{H}}(\boldsymbol{c}_i, \boldsymbol{c}_j)\Big),\end{aligned}$$

where the third equality follows from

$$d_{\mathcal{H}}(\boldsymbol{b},\boldsymbol{c}) = \frac{1}{4m}\big(\|\boldsymbol{b}\|_2^2 + \|\boldsymbol{c}\|_2^2 - 2\langle \boldsymbol{b},\boldsymbol{c}\rangle\big)\ \forall\ \boldsymbol{b},\boldsymbol{c} \in \{-1,1\}^m$$

By using (5.3) to (5.5), we have

$$\big|\boldsymbol{M}_{i,j}\big| \le 8\delta m.$$

Therefore, $\frac{1}{4m\cdot(\mathcal{L}(1)+\delta)}\mathbf{M}$ is actually a $\left(\frac{2\delta}{\mathcal{L}(1)}, \frac{2\delta}{\mathcal{L}(1)}\right)$-near identity matrix. Consider degree k polynomial $P(z) = z^k$. Let

$$\mathbf{N} = P\big(\frac{1}{4m\cdot\mathcal{L}(1)}\mathbf{M}\big).$$

It is easy to observe that $\mathbf{N}$ is a (γ_1,γ_2)-near identity matrix where

$$\gamma_1 = 1 - (1 - \frac{2\delta}{\mathcal{L}(1)})^k,$$

and

$$\gamma_2 = (\frac{2\delta}{\mathcal{L}(1)})^k.$$

Under the condition $\frac{\delta}{\mathcal{L}(1)} \le \frac{1}{4}$, we have

$$\gamma_1 = 1 - (1 - \frac{\delta}{\mathcal{L}(1)})^k \le 1 - (\frac{1}{2})^k.$$

By setting $k = \frac{1}{2}\frac{\log N}{\log\frac{\mathcal{L}(1)}{2\delta}}$, we have

$$\gamma_2 \le \sqrt{\frac{1}{N}}.$$

We apply Lemma 5.4 by setting δ_1, δ_2, p in the statement to be γ_1, γ_2, N respectively. We get

$$\operatorname{rank}(\mathbf{N}) \ge \frac{N(\frac{1}{4})^k}{1 + (N-1)/N} \ge \frac{1}{2}(\frac{1}{4})^k N \ge (\frac{1}{8})^k N. \tag{5.6}$$

On the other hand, $\frac{1}{4m\cdot\mathcal{L}(1)}\mathbf{M}$ has rank at most m. By applying Lemma 5.5 we get

$$\operatorname{rank}(\mathbf{N}) \le \binom{m+k}{k} \le (\frac{e(m+k)}{k})^k.$$

Applying the above result and (5.6) directly yields that

$$(N)^{1/k} \le 8e\frac{m+k}{k}.$$

When $k = \frac{1}{2}\frac{\log N}{\log\frac{\mathcal{L}(1)}{2\delta}}$ as we set, $N^{1/k} \ge (\frac{\mathcal{L}(1)}{2\delta})^2$. Therefore we have

$$m \ge \frac{1}{32e}\big(\frac{\mathcal{L}(1)}{\delta}\big)^2 k - k \ge \frac{1}{64e}\big(\frac{\mathcal{L}(1)}{2\delta}\big)^2 k = \frac{1}{128e}\big(\frac{\mathcal{L}(1)}{\delta}\big)^2 \frac{\log N}{\log\frac{\mathcal{L}(1)}{2\delta}},$$

where the second inequality holds when $\big(\frac{\mathcal{L}(1)}{2\delta}\big)^2 \ge 64e$. □

5.3 Proofs about Fast Binary Embedding Algorithm

5.3.1 Proof of Lemma 3.7

Proof. It suffices to prove $X \perp Y'$. One can check similarly that the proof holds for the remaining three results. Note that X, Y' are binary random variables with values $\{-1, 1\}$. It is easy to observe both of them are balanced, namely $\Pr(X = 1) = \Pr(Y' = 1) = 1/2$. If $X \perp Y'$, then we have $\Pr(X = Y') = 1/2$. In the reverse direction, suppose $\Pr(X = Y') = 1/2$. First we have

$$\Pr(X = 1) = \Pr(X = 1, Y' = 1) + \Pr(X = 1, Y' = -1) = 1/2, \tag{5.7}$$

$$\Pr(Y' = 1) = \Pr(X = 1, Y' = 1) + \Pr(X = -1, Y' = 1) = 1/2. \tag{5.8}$$

Combining the above two results, we have $\Pr(X = 1, Y' = -1) = \Pr(X = -1, Y' = 1)$. Using $\Pr(X = 1, Y' = -1) + \Pr(X = -1, Y' = 1) = \Pr(X \neq Y') = 1 - \Pr(X = Y') = \frac{1}{2}$, we thus have $\Pr(X = 1, Y' = -1) = \Pr(X = -1, Y' = 1) = 1/4$. Plugging the above result into (5.7) and (5.8) we have $\Pr(X = 1, Y' = 1) = \Pr(X = -1, Y' = -1) = 1/4$. Thus we have shown

$$\Pr(X = v \big| Y' = u) = \frac{\Pr(X = v, Y' = u)}{\Pr(Y' = u)} = \Pr(X = v), \ \forall\, u, v \in \{-1, 1\},$$

which leads to $X \perp Y'$.

Using the above arguments, we show that $X \perp Y'$ if and only if

$$\Pr(X = Y') = 1/2.$$

Recalling the definition of X, Y', the above condition holds if and only if

$$\Pr\Big\{ \underbrace{\langle \boldsymbol{\xi} \odot \boldsymbol{\zeta}, \boldsymbol{x} \rangle \cdot \langle \boldsymbol{\xi}' \odot \boldsymbol{\zeta}, \boldsymbol{y} \rangle}_{Z} \ \geq \ 0 \Big\} = \frac{1}{2}.$$

Next we prove Z has symmetric distribution around 0. Let $\mathcal{I} = [1, n], \mathcal{I}' = [1, n - \Delta], \mathcal{I}_0 = [2n - \Delta, 2n - 1]$ for some natural number $\Delta < n$. Without loss of generality, we assume $\boldsymbol{\xi} = \boldsymbol{g}_{\mathcal{I}}$ and $\boldsymbol{\xi}' = [\boldsymbol{g}_{\mathcal{I}_0}; \boldsymbol{g}_{\mathcal{I}'}]$. We split $\mathcal{I}$ into $T = \lceil \frac{n}{\Delta} \rceil$ consecutive disjoint subsets $\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_T$ each of which has size Δ except $|\mathcal{I}_T| = n - (T - 1)\Delta \leq \Delta$. Also, let $\mathcal{I}'_{T-1}$ contain the first $n - (T - 1)\Delta$ entries of $\mathcal{I}_{T-1}$. Then we have

$$Z = \Bigg(\sum_{i=1}^{T} \langle \boldsymbol{g}_{\mathcal{I}_i} \odot \boldsymbol{\zeta}_{\mathcal{I}_i}, \boldsymbol{x}_{\mathcal{I}_i} \rangle \Bigg) \cdot \Bigg(\sum_{i=1}^{T-2} \langle \boldsymbol{g}_{\mathcal{I}_i} \odot \boldsymbol{\zeta}_{\mathcal{I}_{i+1}}, \boldsymbol{y}_{\mathcal{I}_{i+1}} \rangle + \langle \boldsymbol{g}_{\mathcal{I}'_{T-1}} \odot \boldsymbol{\zeta}_{\mathcal{I}_T}, \boldsymbol{y}_{\mathcal{I}_T} \rangle + \langle \boldsymbol{g}_{\mathcal{I}_0} \odot \boldsymbol{\zeta}_{\mathcal{I}_1}, \boldsymbol{y}_{\mathcal{I}_1} \rangle \Bigg). \tag{5.9}$$

We now let $\widehat{\boldsymbol{g}}$ be such random vector that is identical to $\boldsymbol{g}$ except that for any $i \in \{0\} \cup [T]$

$$\widehat{\boldsymbol{g}_{\mathcal{I}_i}} = -\boldsymbol{g}_{\mathcal{I}_i}, \text{ if } i \bmod 2 = 0$$

Let $\widehat{\boldsymbol{\zeta}}$ be such random vector that is identical to $\boldsymbol{\zeta}$ except that for any $i \in \{0\} \cup [T]$

$$\widehat{\boldsymbol{\zeta}_{\mathcal{I}_i}} = -\boldsymbol{\zeta}_{\mathcal{I}_i}, \text{ if } i \bmod 2 = 1.$$

Replacing $\boldsymbol{g},\ \boldsymbol{\zeta}$ in (5.9) with $\widehat{\boldsymbol{g}},\ \widehat{\boldsymbol{\zeta}}$ yields

$$
\begin{aligned}
&\widehat{Z}\\
=&\Big(\sum_{i=1}^{T}\big\langle \widehat{\boldsymbol{g}}_{\mathcal{I}_i}\odot\widehat{\boldsymbol{\zeta}}_{\mathcal{I}_i},\boldsymbol{x}_{\mathcal{I}_i}\big\rangle\Big)\cdot\Big(\sum_{i=1}^{T-2}\big\langle \widehat{\boldsymbol{g}}_{\mathcal{I}_i}\odot\widehat{\boldsymbol{\zeta}}_{\mathcal{I}_{i+1}},\boldsymbol{y}_{\mathcal{I}_{i+1}}\big\rangle+\big\langle \widehat{\boldsymbol{g}}_{\mathcal{I}'_{T-1}}\odot\widehat{\boldsymbol{\zeta}}_{\mathcal{I}_T},\boldsymbol{y}_{\mathcal{I}_T}\big\rangle+\big\langle \widehat{\boldsymbol{g}}_{\mathcal{I}_0}\odot\widehat{\boldsymbol{\zeta}}_{\mathcal{I}_1},\boldsymbol{y}_{\mathcal{I}_1}\big\rangle\Big)\\
=&\Big(-\sum_{i=1}^{T}\big\langle \boldsymbol{g}_{\mathcal{I}_i}\odot\boldsymbol{\zeta}_{\mathcal{I}_i},\boldsymbol{x}_{\mathcal{I}_i}\big\rangle\Big)\cdot\Big(\sum_{i=1}^{T-2}\big\langle \boldsymbol{g}_{\mathcal{I}_i}\odot\boldsymbol{\zeta}_{\mathcal{I}_{i+1}},\boldsymbol{y}_{\mathcal{I}_{i+1}}\big\rangle+\big\langle \boldsymbol{g}_{\mathcal{I}'_{T-1}}\odot\boldsymbol{\zeta}_{\mathcal{I}_T},\boldsymbol{y}_{\mathcal{I}_T}\big\rangle+\big\langle \boldsymbol{g}_{\mathcal{I}_0}\odot\boldsymbol{\zeta}_{\mathcal{I}_1},\boldsymbol{y}_{\mathcal{I}_1}\big\rangle\Big)\\
=&-Z.
\end{aligned}
$$

As each entry of $\boldsymbol{g}$ is symmetric random variable around 0, therefore $\widehat{\boldsymbol{g}}$ and $\boldsymbol{g}$ has the same probability distribution. The same fact also holds for $\widehat{\boldsymbol{\zeta}}$ and $\boldsymbol{\zeta}$. So we conclude that Z has symmetric distribution around 0, which implies $\Pr(Z>0)=\frac{1}{2}$ and $X\perp Y'$. □

5.3.2 Proof of Theorem 3.8

Proof. Unspecified notations in this section are consistent with Algorithm 2. Using Lemma 3.6, we have

$$\Pr\left\{\sup_{i,j\in[N]}\big|d(\boldsymbol{y}_i,\boldsymbol{y}_j)-d(\boldsymbol{x}_i,\boldsymbol{x}_j)\big|\geq C\delta\right\}\leq 0.01. \tag{5.10}$$

Now consider the first-block binary codes generated from Gaussian Toeplitz projection. We focus on two intermediate points $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$. Consider the first block of binary codes generated from the second part of Algorithm 2. We let

$$\boldsymbol{u}=\operatorname{sign}\big(\Psi^{(1)}\cdot\boldsymbol{y}_1\big),\boldsymbol{v}=\operatorname{sign}\big(\Psi^{(1)}\cdot\boldsymbol{y}_2\big).$$

Suppose $\Psi^{(1)}$ contains Gaussian Toeplitz matrix $\mathbf{T}$. For any $i\in[m/B]$, we have

$$u_i=\operatorname{sign}\big(\langle\mathbf{T}_i\odot\boldsymbol{\zeta},\ \boldsymbol{y}_1\rangle\big)=\operatorname{sign}\big(\langle\mathbf{T}_i,\ \boldsymbol{y}_1\odot\boldsymbol{\zeta}\rangle\big).$$

$$v_i=\operatorname{sign}\big(\langle\mathbf{T}_i\odot\boldsymbol{\zeta},\ \boldsymbol{y}_2\rangle\big)=\operatorname{sign}\big(\langle\mathbf{T}_i,\ \boldsymbol{y}_2\odot\boldsymbol{\zeta}\rangle\big).$$

Since $\mathbf{T}_i$ is a Gaussian random vector, we have

$$\Pr(u_i\neq v_i)=d(\boldsymbol{y}_1\odot\boldsymbol{\zeta},\ \boldsymbol{y}_2\odot\boldsymbol{\zeta})=d(\boldsymbol{y}_1,\ \boldsymbol{y}_2).$$

Let $Z_i=\mathbb{1}\big(u_i\neq v_i\big),\forall\ i\in[m/B]$. Following Lemma (3.7), we know that $\forall\ i\neq j$

$$u_i\perp u_j,\ u_i\perp v_j,\ v_i\perp v_j,\ v_i\perp u_j.$$

Therefore $\{Z_i\}_{i=1}^{[m/B]}$ is a pair-wise independent sequence. By Markov's inequality, we have

$$\Pr\left(\Big|\frac{1}{m/B}\sum_{i=1}^{m/B}Z_i-\mathbb{E}(Z_1)\Big|\geq\delta\right)\leq\frac{\frac{B}{m}Var(Z_1)}{\delta^2}\leq\frac{1}{4}\frac{B}{m\delta^2}\leq\frac{1}{4}. \tag{5.11}$$

The last inequality holds by setting $\frac{m}{B} \geq \frac{1}{\delta^2}$. Therefore, we have

$$\Pr\left(\left|d_{\mathcal{H}}(\boldsymbol{u},\boldsymbol{v}) - d(\boldsymbol{y}_1,\boldsymbol{y}_2)\right| \geq \delta\right) \leq \frac{1}{4}.$$

Now consider total B block binary codes $\{\boldsymbol{u}_i\}_{i=1}^B$ $\{\boldsymbol{v}_i\}_{i=1}^B$ from $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$ respectively. Let

$$E_i = \mathbb{1}\big(|d_{\mathcal{H}}(\boldsymbol{u}_i,\boldsymbol{v}_i) - d(\boldsymbol{y}_1,\boldsymbol{y}_2)| \geq \delta\big),\ \forall\, i \in [B].$$

From (5.11), we have $\Pr(E_i = 1) < \frac{1}{4}$. If more than half of E_i are 0, then the median of $\{d_{\mathcal{H}}(\boldsymbol{u}_i,\boldsymbol{v}_i)\}_{i=1}^B$ is within δ away from $d(\boldsymbol{y}_1,\boldsymbol{y}_2)$. Then we have

$$\begin{aligned}
&\Pr\left(\text{median}\left(\{d_{\mathcal{H}}(\boldsymbol{u}_i,\boldsymbol{v}_i)\}_{i=1}^B\right) - d(\boldsymbol{y}_1,\boldsymbol{y}_2)\right| \geq \delta\Big) \\
&\leq \Pr\big(\frac{1}{B}\sum_{i=1}^B E_i \geq \frac{1}{2}\big) \leq \Pr\big(\frac{1}{B}\sum_{i=1}^B E_i - \mathbb{E}(E_i) > \frac{1}{4}\big) \leq \exp(-\frac{1}{4}B).
\end{aligned}$$

In the second inequality, we use (5.11). The last step follows from Hoeffding's inequality. Now we use a union bound for N^2 pairs

$$\Pr\left(\sup_{i,j\in[N]}\left|d_{\mathcal{H}}(\boldsymbol{b}_i,\boldsymbol{b}_j) - d(\boldsymbol{y}_i,\boldsymbol{y}_j)\right| \geq \delta\right) \leq N^2\exp(-\frac{1}{4}B) \leq \exp(-\frac{1}{8}B).$$

The last inequality holds by setting $B \geq 16\log N$. Combing the above result and (5.10) using triangle inequality, we complete the proof. □

5.4 Proof of Theorem 3.10

For any set $K \subseteq \mathbb{S}^{p-1}$, we use $\mathcal{N}_\delta(K)$ to denote a constructed δ-net of K, which is a δ-covering set with minimum size. In particular, by Sudakov's theorem (e.g., Theorem 3.18 in Ledoux and Talagrand (1991))

$$\log \mathcal{N}_\delta(K) \lesssim \frac{w(K)^2}{\delta^2}.$$

We first prove that for a fixed two dimensional space, $m = O(\frac{1}{\delta^2})$ independent Gaussian measurements are sufficient to achieve δ-uniform binary embedding.

Lemma 5.6. Suppose K is any fixed two-dimensional subspace in $\mathbb{S}^{p-1}$. Let $\mathbf{A} \in \mathbb{R}^{m\times p}$ be a matrix with independent rows $\mathbf{A}_i \sim \mathcal{N}(0,\mathbf{I}_p)$, $\forall i \in [m]$. Suppose $m \geq \frac{1}{\delta^2}\log\frac{1}{\delta}$, then with probability at least $1 - 3\exp(-\delta^2 m)$,

$$\sup_{\boldsymbol{x},\boldsymbol{y}\in K}\left|d_{\mathbf{A}}(\boldsymbol{x},\boldsymbol{y}) - d(\boldsymbol{x},\boldsymbol{y})\right| \leq C\delta. \tag{5.12}$$

Here C is some absolute constant.

Proof. We postpone the proof to Appendix C. □

The next lemma shows that the normalized ℓ_1 norm of $\mathbf{A}\boldsymbol{x}$ provides decent approximation of $\|\boldsymbol{x}\|_2$.

Lemma 5.7. Consider any set $K \subseteq \mathbb{R}^p$. Let $\mathbf{A}$ be an m-by-p matrix with independent rows $\mathbf{A}_i \sim \mathcal{N}(0, \mathbf{I}_p)$ for any $i \in [m]$. Consider

$$Z = \sup_{\boldsymbol{x} \in K} \left| \frac{1}{m} \sum_{i=1}^{m} \left| \langle \mathbf{A}_i, \boldsymbol{x} \rangle \right| - \sqrt{\frac{2}{\pi}} \|\boldsymbol{x}\|_2 \right|.$$

We have

$$\Pr\left\{ Z \geq 4\frac{w(K)}{\sqrt{m}} + t \right\} \leq 2\exp\left(-\frac{mt^2}{2d(K)^2} \right), \ \forall\, t > 0.$$

where $d(K) = \max_{\boldsymbol{x} \in K} \|\boldsymbol{x}\|_2$.

Proof. See the proof of Lemma 2.1 in Plan and Vershynin (2014). □

In order to connect ℓ_1 norm to Hamming distance, we need the following result.

Lemma 5.8. Consider finite number of points $K \subseteq \mathbb{S}^{p-1}$. Let $\mathbf{A}$ be an m-by-p matrix with independent rows $\mathbf{A}_i \sim \mathcal{N}(0, \mathbf{I}_p)$ for any $i \in [m]$. Suppose

$$m \geq \frac{1}{\delta^2} \log |K|,$$

then we have

$$\sup_{\boldsymbol{x} \in |K|} \frac{1}{m} \sum_{i=1}^{m} \mathbb{1}\left\{ \left| \langle \mathbf{A}_i, \boldsymbol{x} \rangle \right| \leq \delta \right\} \leq 2\delta.$$

with probability at least $1 - \exp(-\delta^2 m)$.

Proof. Let $X \sim \mathcal{N}(0, 1)$. For any fixed point $\boldsymbol{x} \in K$ and any $i \in [m]$, we have

$$\Pr(\left| \langle \boldsymbol{A}_i, \boldsymbol{x} \rangle \right| \leq \delta) = \Pr(|X| \leq \delta) \leq \delta.$$

Let $Z_i = \mathbb{1}(\left| \langle \boldsymbol{A}_i, \boldsymbol{x} \rangle \right| \leq \delta)$, $\forall\, i \in [m]$. Then by using Hoeffding's inequality,

$$\Pr(\frac{1}{m} \sum_{i=1}^{m} Z_i - \mathbb{E}(Z_1) > \delta) \leq \exp(-2\delta^2 m).$$

As $\mathbb{E}(Z_1) = \Pr(\left| \langle \boldsymbol{A}_i, \boldsymbol{x} \rangle \right| \leq \delta) \leq \delta$, we conclude that with probability at least $1 - \exp(-2\delta^2 m)$,

$$\frac{1}{m} \sum_{i=1}^{m} Z_i \leq 2\delta.$$

By applying union bound over $|K|$ points and setting $m \geq \frac{1}{\delta^2} \log |K|$, we complete the proof. □

Now we are ready to prove Theorem 3.10.

Proof of Theorem 3.10. We construct a δ-net of K that is denoted as $\mathcal{N}_\delta$. We assume $m \gtrsim \frac{1}{\delta^2}\log|\mathcal{N}_\delta|$. Applying Proposition 2.2 and setting $K = \mathcal{N}_\delta$, we have that

$$\sup_{\boldsymbol{x},\boldsymbol{y}\in\mathcal{N}_\delta} \left|d_{\mathbf{A}}(\boldsymbol{x},\boldsymbol{y}) - d(\boldsymbol{x},\boldsymbol{y})\right| \le \delta \tag{5.13}$$

with probability at least $1 - 2\exp(-\delta^2 m)$.

For any two fixed points $\boldsymbol{x}, \boldsymbol{y} \in K$, let $\boldsymbol{x}_1, \boldsymbol{y}_1$ be their nearest points in $\mathcal{N}_\delta$. Then we have

$$\begin{aligned}
&|d(\boldsymbol{x},\boldsymbol{y}) - d_{\mathbf{A}}(\boldsymbol{x},\boldsymbol{y})| \le |d(\boldsymbol{x},\boldsymbol{y}) - d(\boldsymbol{x}_1,\boldsymbol{y}_1)| + |d(\boldsymbol{x}_1,\boldsymbol{y}_1) - d_{\mathbf{A}}(\boldsymbol{x},\boldsymbol{y})| \\
&\overset{(a)}{\le} |d(\boldsymbol{x}_1,\boldsymbol{y}_1) - d_{\mathbf{A}}(\boldsymbol{x},\boldsymbol{y})| + 2\delta \le |d_{\mathbf{A}}(\boldsymbol{x}_1,\boldsymbol{y}_1) - d_{\mathbf{A}}(\boldsymbol{x},\boldsymbol{y})| + |d(\boldsymbol{x}_1,\boldsymbol{y}_1) - d_{\mathbf{A}}(\boldsymbol{x}_1,\boldsymbol{y}_1)| + 2\delta \\
&\overset{(b)}{\le} |d_{\mathbf{A}}(\boldsymbol{x}_1,\boldsymbol{y}_1) - d_{\mathbf{A}}(\boldsymbol{x},\boldsymbol{y})| + 3\delta \le |d_{\mathbf{A}}(\boldsymbol{x}_1,\boldsymbol{y}_1) - d_{\mathbf{A}}(\boldsymbol{x}_1,\boldsymbol{y})| + |d_{\mathbf{A}}(\boldsymbol{x}_1,\boldsymbol{y}) - d_{\mathbf{A}}(\boldsymbol{x},\boldsymbol{y})| + 3\delta \\
&\overset{(c)}{\le} d_{\mathbf{A}}(\boldsymbol{y}_1,\boldsymbol{y}) + d_{\mathbf{A}}(\boldsymbol{x}_1,\boldsymbol{x}) + 3\delta,
\end{aligned} \tag{5.14}$$

where (a) follows from

$$|d(\boldsymbol{x},\boldsymbol{y}) - d(\boldsymbol{x}_1,\boldsymbol{y}_1)| \le |d(\boldsymbol{x},\boldsymbol{y}) - d(\boldsymbol{x}_1,\boldsymbol{y})| + |d(\boldsymbol{x}_1,\boldsymbol{y}) - d(\boldsymbol{x},\boldsymbol{y}_1)| \le d(\boldsymbol{x},\boldsymbol{x}_1) + d(\boldsymbol{x}_1,\boldsymbol{y}_1) \le 2\delta,$$

step (b) follows from (5.13), step (c) follows from the triangle inequality of Hamming distance. Therefore we have

$$\sup_{\boldsymbol{x},\boldsymbol{y}\in K} \left|d_{\mathbf{A}}(\boldsymbol{x},\boldsymbol{y}) - d(\boldsymbol{x},\boldsymbol{y})\right| \le 2 \sup_{\boldsymbol{x}_1\in\mathcal{N}_\delta} \sup_{\substack{\boldsymbol{x}\in K \\ \|\boldsymbol{x}-\boldsymbol{x}_1\|_2\le\delta}} d_{\mathbf{A}}(\boldsymbol{x},\boldsymbol{x}_1) + 3\delta. \tag{5.15}$$

Next we bound the tail term

$$T := \sup_{\boldsymbol{x}_1\in\mathcal{N}_\delta} \sup_{\substack{\boldsymbol{x}\in K \\ \|\boldsymbol{x}-\boldsymbol{x}_1\|_2\le\delta}} d_{\mathbf{A}}(\boldsymbol{x},\boldsymbol{x}_1).$$

Recall that

$$K_\delta^+ := K \bigcup \left\{\boldsymbol{z} \in \mathbb{S}^{p-1} : \boldsymbol{z} = \frac{\boldsymbol{x}-\boldsymbol{y}}{\|\boldsymbol{x}-\boldsymbol{y}\|_2},\ \forall\, \boldsymbol{x},\boldsymbol{y} \in K \text{ if } \delta^2 \le \|\boldsymbol{x}-\boldsymbol{y}\|_2 \le \delta\right\}.$$

Now we construct a δ-net for $K_\delta^+ \setminus K$ denoted as $\mathcal{N}_\delta'$. For two distinct points $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{N}_\delta' \bigcup \mathcal{N}_\delta$, let $C(\boldsymbol{x},\boldsymbol{y})$ denote the unit circle spanned by $\boldsymbol{x}, \boldsymbol{y}$. We construct δ^2-net $\mathcal{C}_{\delta^2}(\boldsymbol{x},\boldsymbol{y})$ for each circle $\mathcal{C}(\boldsymbol{x},\boldsymbol{y})$. For simplicity, we just let $\mathcal{C}_{\delta^2}(\boldsymbol{x},\boldsymbol{y})$ be the set of points that uniformly split $\mathcal{C}(\boldsymbol{x},\boldsymbol{y})$ with interval δ^2. We thus have $|\mathcal{C}_{\delta^2}(\boldsymbol{x},\boldsymbol{y})| \lesssim \frac{1}{\delta^2}$. Let $\mathcal{G}_\delta$ denote the union of all circle nets $\mathcal{C}_{\delta^2}(\boldsymbol{x},\boldsymbol{y})$ spanned by points in $\mathcal{N}_\delta' \bigcup \mathcal{N}_\delta$, namely

$$\mathcal{G}_\delta := \bigcup_{\forall\, \boldsymbol{x},\boldsymbol{y}\in\mathcal{N}_\delta' \bigcup \mathcal{N}_\delta} \mathcal{C}_{\delta^2}(\boldsymbol{x},\boldsymbol{y}) \cup \{\boldsymbol{x},\boldsymbol{y}\}.$$

For any point $\boldsymbol{x} \in K$, we can always find a point in $\mathcal{G}_\delta$ that is $O(\delta^2)$ away from $\boldsymbol{x}$. To see why the argument is true, we first let $\boldsymbol{x}_1$ be the nearest point to $\boldsymbol{x}$ in $\mathcal{N}_\delta$. If $\|\boldsymbol{x}-\boldsymbol{x}_1\|_2 \le \delta^2$,

then $\boldsymbol{x}_1$ is the point we want. Otherwise, we have $\delta^2 \le \|\boldsymbol{x}-\boldsymbol{x}_1\|_2 \le \delta$. In this case, we have $(\boldsymbol{x}-\boldsymbol{x}_1)/\|\boldsymbol{x}-\boldsymbol{x}_1\| \in K^+$. Following the definition of K_δ^+, we can always find a point $\boldsymbol{x}_1' \in \mathcal{N}_\delta' \bigcup \mathcal{N}_\delta$ such that

$$\big\|\boldsymbol{x}_1' - \frac{\boldsymbol{x}-\boldsymbol{x}_1}{\|\boldsymbol{x}-\boldsymbol{x}_1\|_2}\big\|_2 \le \delta, \tag{5.16}$$

thereby

$$\big\|\boldsymbol{x} - \underbrace{\big(\|\boldsymbol{x}-\boldsymbol{x}_1\|_2\boldsymbol{x}_1' + \boldsymbol{x}_1\big)}_{\boldsymbol{z}}\big\|_2 \le \delta\|\boldsymbol{x}-\boldsymbol{x}_1\|_2 \le \delta^2.$$

Note that $\|\boldsymbol{z}\|_2$ is very close to 1 because

$$\delta^4 \ge \|\boldsymbol{x}-\boldsymbol{z}\|_2^2 \ge \|\boldsymbol{z}\|_2^2 - 2\langle \boldsymbol{z}, \boldsymbol{x}\rangle + 1 \ge \|\boldsymbol{z}\|_2^2 - 2\|\boldsymbol{z}\|_2 + 1 = (\|\boldsymbol{z}\|_2 - 1)^2.$$

We thus have

$$\big\|\boldsymbol{x} - \boldsymbol{z}/\|\boldsymbol{z}\|_2\big\|_2 \le \|\boldsymbol{x}-\boldsymbol{z}\|_2 + \big\|\boldsymbol{z} - \boldsymbol{z}/\|\boldsymbol{z}\|_2\big\|_2 = \|\boldsymbol{x}-\boldsymbol{z}\|_2 + \big|\|\boldsymbol{z}\|_2 - 1\big| \le 2\delta^2.$$

Note that $\boldsymbol{z}$ is in the unit circle $\mathcal{C}(\boldsymbol{x}, \boldsymbol{x}_1')$ spanned by $\boldsymbol{x}$ and $\boldsymbol{x}_1'$, thereby there exists $\boldsymbol{u} \in \mathcal{C}_{\delta^2}(\boldsymbol{x}_1, \boldsymbol{x}_1')$ such that $\|\boldsymbol{u}-\boldsymbol{x}\|_2 \le \delta^2$. Point $\boldsymbol{u}$ thus satisfies

$$\|\boldsymbol{x}-\boldsymbol{u}\| \le \|\boldsymbol{x}-\boldsymbol{z}\|_2 + \|\boldsymbol{z}-\boldsymbol{u}\|_2 \le 3\delta^2. \tag{5.17}$$

So for any $\boldsymbol{x} \in K$ and its nearest point $\boldsymbol{x}_1 \in \mathcal{N}_\delta$, we define $\boldsymbol{u}$ as

$$\boldsymbol{u} := \begin{cases} \boldsymbol{x}_1, & \|\boldsymbol{x}-\boldsymbol{x}_1\|_2 \le \delta^2; \\ \operatorname{argmin}_{\boldsymbol{v}\in\mathcal{C}_{\delta^2}(\boldsymbol{x}_1,\boldsymbol{x}_1')} \|\boldsymbol{x}-\boldsymbol{v}\|_2, & \text{otherwise.} \end{cases}$$

where $\boldsymbol{x}_1' \in \mathcal{N}_\delta \bigcup \mathcal{N}_\delta'$ and satisfies (5.16). Based on (5.17), we always have $\|\boldsymbol{u}-\boldsymbol{x}\|_2 \le 3\delta^2$ and $\|\boldsymbol{u}-\boldsymbol{x}_1\|_2 \le \|\boldsymbol{u}-\boldsymbol{x}\|_2 + \|\boldsymbol{x}-\boldsymbol{x}_1\|_2 \le 2\delta$.

By triangle inequality of Hamming distance,

$$d_{\mathbf{A}}(\boldsymbol{x}, \boldsymbol{x}_1) \le d_{\mathbf{A}}(\boldsymbol{x}, \boldsymbol{u}) + d_{\mathbf{A}}(\boldsymbol{u}, \boldsymbol{x}_1).$$

We thus have

$$\begin{aligned} T \le & \sup_{\boldsymbol{x}_1\in\mathcal{N}_\delta} \sup_{\substack{\boldsymbol{x}\in K \\ \|\boldsymbol{x}-\boldsymbol{x}_1\|_2}} d_{\mathbf{A}}(\boldsymbol{x}, \boldsymbol{u}) + d_{\mathbf{A}}(\boldsymbol{u}, \boldsymbol{x}_1) \\ \le & \underbrace{\sup_{\boldsymbol{u}\in\mathcal{G}_\delta} \sup_{\substack{\boldsymbol{x}\in K \\ \|\boldsymbol{x}-\boldsymbol{u}\|_2\le 3\delta^2}} d_{\mathbf{A}}(\boldsymbol{x}, \boldsymbol{u})}_{T_1} + \underbrace{\sup_{\boldsymbol{x},\boldsymbol{y}\in\mathcal{N}_\delta\bigcup\mathcal{N}_\delta'} \sup_{\substack{\boldsymbol{u},\boldsymbol{v}\in\mathcal{C}(\boldsymbol{x},\boldsymbol{y}) \\ \|\boldsymbol{u}-\boldsymbol{v}\|_2\le 2\delta}} d_{\mathbf{A}}(\boldsymbol{u}, \boldsymbol{v})}_{T_2}. \end{aligned}$$

Next we bound term T_1 and T_2 respectively.

Term T_1. For a fixed point $\boldsymbol{u} \in \mathcal{G}_\delta$, using Lemma 5.7 by setting (K, t) in the statement to be $K' = (K - \{\boldsymbol{u}\}) \bigcap \{\boldsymbol{u} \in \mathbb{R}^p : \|\boldsymbol{u}\|_2 \le 3\delta^2\}$ and δ^2 respectively yields that

$$\begin{aligned} &\Pr\Big\{ \sup_{\substack{\boldsymbol{x}\in K \\ \|\boldsymbol{x}-\boldsymbol{u}\|_2\le 3\delta^2}} \Big| \frac{1}{m}\sum_{i=1}^m \big|\langle \mathbf{A}_i, \boldsymbol{x}-\boldsymbol{u}\rangle\big| - \sqrt{\frac{2}{\pi}}\|\boldsymbol{x}-\boldsymbol{u}\|_2 \Big| \ge \frac{4w(K')}{\sqrt{m}} + \delta^2 \Big\} \\ \le & 2\exp\big(-\frac{m\delta^4}{2d(K')^2}\big) \le 2\exp(-m/18). \end{aligned}$$

Then with probability greater than $1-2\exp(-m/18)$,

$$\sup_{\substack{\boldsymbol{x}\in K\\ \|\boldsymbol{x}-\boldsymbol{u}\|_2\le 3\delta^2}} \frac{1}{m}\sum_{i=1}^{m}\left|\langle \mathbf{A}_i, \boldsymbol{x}-\boldsymbol{u}\rangle\right| \le 3\sqrt{\frac{2}{\pi}}\delta^2 + 4w(K')/\sqrt{m} + \delta^2 \le 5\delta^2,$$

where the last inequality follows from the fact that $w(K') \lesssim w(K)$ and our assumption $m \gtrsim w(K)^2/\delta^4$. We define event

$$\mathcal{E} := \left\{ \sup_{\boldsymbol{u}\in\mathcal{G}_\delta} \sup_{\substack{\boldsymbol{x}\in K\\ \|\boldsymbol{x}-\boldsymbol{u}\|_2\le 3\delta^2}} \frac{1}{m}\sum_{i=1}^{m}\left|\langle \mathbf{A}_i, \boldsymbol{x}-\boldsymbol{u}\rangle\right| \le 5\delta^2 \right\}.$$

Applying union bound over all points in $\mathcal{G}_\delta$, we have

$$\Pr(\mathcal{E}^c) \le 2|\mathcal{G}_\delta|\exp(-m/18) \le 2\exp(-m/36),$$

where the last inequality holds with $m \gtrsim \log|\mathcal{G}_\delta|$. Under condition event $\mathcal{E}$ happens, we have

$$\sup_{\boldsymbol{u}\in\mathcal{G}_\delta} \sup_{\substack{\boldsymbol{x}\in K\\ \|\boldsymbol{x}-\boldsymbol{u}\|_2\le 3\delta^2}} \frac{1}{m}\sum_{i=1}^{m}\mathbb{1}\left\{\left|\langle \mathbf{A}_i, \boldsymbol{u}-\boldsymbol{x}\rangle\right| \le 5\delta\right\} \ge 1-\delta. \tag{5.18}$$

If $\operatorname{sign}\left(\langle \mathbf{A}_i, \boldsymbol{u}\rangle\right) \ne \operatorname{sign}\left(\langle \mathbf{A}_i, \boldsymbol{x}\rangle\right)$, we must have $\left|\langle \mathbf{A}_i, \boldsymbol{u}\rangle\right| \le \left|\langle \mathbf{A}_i, \boldsymbol{u}-\boldsymbol{x}\rangle\right|$. We then have

$$\begin{aligned} T_1 &\le \sup_{\boldsymbol{u}\in\mathcal{G}_\delta} \sup_{\substack{\boldsymbol{x}\in K\\ \|\boldsymbol{x}-\boldsymbol{u}\|_2\le 3\delta^2}} \frac{1}{m}\sum_{i=1}^{m}\mathbb{1}\left\{\left|\langle \mathbf{A}_i, \boldsymbol{u}\rangle\right| \le \left|\langle \mathbf{A}_i, \boldsymbol{u}-\boldsymbol{x}\rangle\right|\right\} \\ &\le \sup_{\boldsymbol{u}\in\mathcal{G}_\delta} \frac{1}{m}\sum_{i=1}^{m}\mathbb{1}\left\{\left|\langle \mathbf{A}_i, \boldsymbol{u}\rangle\right| \le 5\delta\right\} + \delta, \end{aligned}$$

where the last inequality follows from (5.18). Using Lemma 5.7 by setting K and δ in the statement to be $\mathcal{G}_\delta$ and 5δ respectively, we have that, when $m \ge c\frac{1}{\delta^2}\log|\mathcal{G}_\delta|$ with some absolute constant c, the following inequality

$$\sup_{\boldsymbol{u}\in\mathcal{G}_\delta} \frac{1}{m}\sum_{i=1}^{m}\mathbb{1}\left\{\left|\langle \mathbf{A}_i, \boldsymbol{u}\rangle\right| \le 5\delta\right\} \le 10\delta$$

holds with probability at least $1-\exp(-25\delta^2 m)$. Putting all ingredients together, we have $T_1 \le 11\delta$ with high probability.

Term T_2. There are at most $|\mathcal{N}_\delta \bigcup \mathcal{N}'_\delta|^2$ different two-dimensional subspaces constructed from $\mathcal{N}_\delta \bigcup \mathcal{N}'_\delta$. Applying Lemma 5.6 and probabilistic union bound over all subspaces yields that

$$\Pr\left(T_2 \ge (C+2)\delta\right) \le 3\left|\mathcal{N}_\delta \bigcup \mathcal{N}'_\delta\right|^2 \exp(-\delta^2 m) \le 3\exp(-\delta^2 m/2),$$

where the last inequality holds by setting $m \gtrsim \frac{1}{\delta^2}\log|\mathcal{N}_\delta \bigcup \mathcal{N}'_\delta|$.

Putting (5.15) and the upper bounds of term T together, we conclude that by choosing

$$m \gtrsim \max\left\{ w(K)^2/\delta^4,\ \log|\mathcal{G}_\delta|,\ \frac{1}{\delta^2}\log|\mathcal{N}_\delta \bigcup \mathcal{N}'_\delta| \right\},$$

we have

$$\sup_{\boldsymbol{x},\boldsymbol{y}\in K} |d_{\mathbf{A}}(\boldsymbol{x},\boldsymbol{y}) - d(\boldsymbol{x},\boldsymbol{y})| \lesssim \delta.$$

with probability at least $1 - c_1 \exp(-c_2\delta^2 m)$ where c_1, c_2 are some absolute constants.

Using the fact that

$$|\mathcal{G}_\delta| \lesssim \frac{1}{\delta^2}|\mathcal{N}_\delta \bigcup \mathcal{N}'_\delta|$$

and

$$\log|\mathcal{N}_\delta \bigcup \mathcal{N}'_\delta| \lesssim \frac{1}{\delta^2} w(\mathcal{N}_\delta \bigcup \mathcal{N}'_\delta)^2 \leq \frac{1}{\delta^2} w(K_\delta^+)^2,$$

we complete the proof. □

A Proof of Lemma 3.6

Proof. Recall that $\boldsymbol{y}_i = \sqrt{\frac{p}{m}}\Phi(\boldsymbol{\zeta}) \cdot \boldsymbol{x}_i$. We let

$$\widehat{\boldsymbol{y}}_i = \frac{\boldsymbol{y}_i}{\|\boldsymbol{y}_i\|_2}, \widehat{\boldsymbol{y}}_j = \frac{\boldsymbol{y}_j}{\|\boldsymbol{y}_j\|_2}.$$

From condition (3.7), we have

$$\|\boldsymbol{y}_i - \widehat{\boldsymbol{y}}_i\|_2 \leq \delta, \|\boldsymbol{y}_j - \widehat{\boldsymbol{y}}_j\|_2 \leq \delta. \tag{A.1}$$

Let $\theta = \angle(\boldsymbol{x}_i, \boldsymbol{x}_j)$, $\theta' = \angle(\widehat{\boldsymbol{y}}_i, \widehat{\boldsymbol{y}}_j)$. Without loss of generality, we assume our set $K = \{\boldsymbol{x}_i\}_{i=1}^N$ is symmetric, i.e., if $\boldsymbol{x} \in K$ then $-\boldsymbol{x} \in \boldsymbol{K}$. Suppose we show for any two points $\boldsymbol{x}_i, \boldsymbol{x}_j$ with $\langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle > 0$, inequality (3.8) holds, then for $\boldsymbol{x}_i, \boldsymbol{x}_j$ with $\langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle < 0$, we immediately have

$$\big|d(\boldsymbol{y}_i, \boldsymbol{y}_j) - d(\boldsymbol{x}_i, \boldsymbol{x}_j)\big| = \big|1 - d(\boldsymbol{y}_i, \boldsymbol{y}_j) - \big(1 - d(\boldsymbol{x}_i, \boldsymbol{x}_j)\big)\big| = \big|d(-\boldsymbol{y}_i, \boldsymbol{y}_j) - d(-\boldsymbol{x}_i, \boldsymbol{x}_j)\big| \leq C\delta.$$

In the second equality, we use $d(-\boldsymbol{x}, \boldsymbol{y}) + d(\boldsymbol{x}, \boldsymbol{y}) = 1$, $\forall\, \boldsymbol{x}, \boldsymbol{y} \in \mathbb{S}^{p-1}$. In the last inequality, we use the fact that fast JL transform $\sqrt{\frac{p}{m}}\Phi(\boldsymbol{\zeta})$ is linear thus $-\boldsymbol{y}_i = \sqrt{\frac{p}{m}}\Phi(\boldsymbol{\zeta})(-\boldsymbol{x}_i)$. Therefore, without loss of generality, we assume $\langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle \geq 0$ thus $\theta \leq \frac{\pi}{2}$.

Now we turn to the following quantity

$$\begin{aligned}\big\|\widehat{\boldsymbol{y}}_i - \widehat{\boldsymbol{y}}_j\big\|_2 &= \big\|\widehat{\boldsymbol{y}}_i - \boldsymbol{y}_i + \boldsymbol{y}_i - \boldsymbol{y}_j + \boldsymbol{y}_j - \widehat{\boldsymbol{y}}_j\big\|_2 \\ &\leq \big\|\widehat{\boldsymbol{y}}_i - \boldsymbol{y}_i\big\|_2 + \big\|\widehat{\boldsymbol{y}}_j - \boldsymbol{y}_j\big\|_2 + \big\|\boldsymbol{y}_i - \boldsymbol{y}_j\big\|_2 \leq 2\delta + \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2(1+\delta).\end{aligned}$$

The last inequality follows from (A.1) and condition (3.6). Similarly, we also have

$$\big\|\widehat{\boldsymbol{y}}_i - \widehat{\boldsymbol{y}}_j\big\|_2 \geq \|\boldsymbol{x}_i - \boldsymbol{x}_j\|(1-\delta) - 2\delta.$$

Using the fact that

$$\sin\frac{\theta'}{2} = \frac{\big\|\widehat{\boldsymbol{y}}_i - \widehat{\boldsymbol{y}}_j\big\|_2}{2}, \sin\frac{\theta}{2} = \frac{\big\|\boldsymbol{x}_i - \boldsymbol{x}_j\big\|_2}{2},$$

we have

$$\big|\sin\frac{\theta'}{2} - \sin\frac{\theta}{2}\big| = \big|\frac{\big\|\widehat{\boldsymbol{y}}_i - \widehat{\boldsymbol{y}}_j\big\|_2}{2} - \frac{\big\|\boldsymbol{x}_i - \boldsymbol{x}_j\big\|_2}{2}\big| \leq \delta + \delta\frac{\big\|\boldsymbol{x}_i - \boldsymbol{x}_j\big\|_2}{2} \leq 2\delta.$$

When $\delta < \frac{\sqrt{3}-\sqrt{2}}{4}$, we have

$$\sin\frac{\theta'}{2} \le \sin\frac{\theta}{2} + \frac{\sqrt{3}-\sqrt{2}}{2} \le \frac{\sqrt{3}}{2}.$$

In the last inequality, we use $\sin\frac{\theta}{2} \le \frac{\sqrt{2}}{2}$, $\forall\ \theta \in [0, \pi/2]$. So $\theta'/2 \in [0, \pi/3]$. Using the fact that, for any two $\theta, \theta' \in [0, \pi/3]$, there exists constant c such that

$$\big|\sin\theta - \sin\theta'\big| \ge c\big|\theta - \theta'\big|,$$

we have that

$$\big|\frac{\theta}{2} - \frac{\theta'}{2}\big| \le \frac{1}{c}\big|\sin\frac{\theta'}{2} - \sin\frac{\theta}{2}\big| \le \frac{2\delta}{c}.$$

Therefore,

$$\big|d(\boldsymbol{y}_i, \boldsymbol{y}_j) - d(\boldsymbol{x}_i, \boldsymbol{x}_j)\big| = \frac{1}{\pi}\big|\theta - \theta'\big| \le C\delta.$$

In the case $\delta > \frac{\sqrt{3}-\sqrt{2}}{4}$, trivially we have $\big|d(\boldsymbol{y}_i, \boldsymbol{y}_j) - d(\boldsymbol{x}_i, \boldsymbol{x}_j)\big| \le 2 \le C\delta$ with constant $C = \frac{8}{\sqrt{3}-\sqrt{2}}$. □

B Proof of Lemma 5.4

Proof. For positive semidefinite matrix $\mathbf{M} \in \mathbb{R}^{p\times p}$ with rank d, let $\lambda_1, \lambda_2, ...\lambda_d$ be its positive eigenvalues. Using the definition of Frobenius norm, we have

$$\|\mathbf{M}\|_F^2 = \sum_{i=1}^{d}\lambda_i^2 = \sum_{i,j\in[n]}(\mathbf{M}_{i,j})^2 \le p + (p^2 - p)\delta_2^2.$$

On the other hand, considering the trace of $\mathbf{M}$, we can obtain

$$\sum_{i=1}^{d}\lambda_i = \mathrm{Trace}(\mathbf{M}) \ge p(1-\delta_1). \tag{B.1}$$

Using Cauchy-Schwarz inequality, we have

$$(\sum_{i=1}^{d}\lambda_i)^2 \le d\sum_{i=1}^{d}\lambda_i^2. \tag{B.2}$$

Plugging (B.1) and (B.2) into the above inequality yields

$$d \ge \frac{p(1-\delta_1)^2}{1 + (p-1)\delta_2^2}.$$

□

C Proof of Lemma 5.12

Proof. Without loss of any generality, we assume $K = \{\boldsymbol{x} \in \mathbb{S}^{p-1} : \mathrm{supp}(\boldsymbol{x}) \subseteq \{1, 2\}\}$. We begin with constructing a δ-net denoted as $\mathcal{N}_\delta$ for set K. For simplicity, we can just let $\mathcal{N}_\delta(K)$ be the set of points that split the circle spanned by $\{\boldsymbol{e}_1, \boldsymbol{e}_2\}$ uniformly. Therefore $|\mathcal{N}_\delta(K)| = O(\frac{1}{\delta})$. Applying Proposition 2.2 gives us

$$\sup_{\boldsymbol{x},\boldsymbol{y}\in\mathcal{N}_\delta} |d_A(\boldsymbol{x}, \boldsymbol{y}) - d(\boldsymbol{x}, \boldsymbol{y})| \leq \delta, \tag{C.1}$$

holds with probability at least $1 - 2\exp(-\delta^2 m)$ when $m \gtrsim \frac{1}{\delta^2}\log(\frac{1}{\delta})$.

For any point $\boldsymbol{x} \in K$, $\langle \mathbf{A}_i, \boldsymbol{x}\rangle$ only depends on the first two coordinates of $\mathbf{A}_i$. Therefore, for simplicity, we let $\mathbf{A}'_i = \frac{\mathbf{A}_i \odot (\boldsymbol{e}_1+\boldsymbol{e}_2)}{\|\mathbf{A}_i \odot (\boldsymbol{e}_1+\boldsymbol{e}_2)\|_2}$, $\forall\ i \in [m]$. For any point say $\boldsymbol{x}_1 \in \mathcal{N}_\delta$, using the uniform distribution of $\mathbf{A}'_i$, we have

$$\Pr(|\langle \mathbf{A}'_i, \boldsymbol{x}_1\rangle| \leq \delta) \lesssim C\delta,$$

holds with some absolute constant C. Using Hoeffding's inequality and probabilistic union bound over all points in $\mathcal{N}_\delta$, we have

$$\Pr\left(\sup_{\boldsymbol{x}\in\mathcal{N}_\delta} \frac{1}{m}\sum_{i=1}^m \mathbb{1}\{|\langle \mathbf{A}_i, \boldsymbol{x}\rangle| \leq \delta\} > (C+1)\delta\right) \leq |\mathcal{N}_\delta|\exp(-2\delta^2 m) \leq \exp(-\delta^2 m). \tag{C.2}$$

The last inequality holds when $m \gtrsim \frac{1}{\delta^2}\log\frac{1}{\delta}$.

Now we consider any point $\boldsymbol{x} \in K$. Suppose $\boldsymbol{x}_1$ is the closest point to $\boldsymbol{x}$ in $\mathcal{N}_\delta$. We note that if $\mathrm{sign}\left(\langle \mathbf{A}'_i, \boldsymbol{x}\rangle\right) \neq \mathrm{sign}\left(\langle \mathbf{A}'_i, \boldsymbol{x}_1\rangle\right)$, then there exists $\lambda \in [0, 1]$ such that

$$\langle \mathbf{A}'_i, \lambda\boldsymbol{x} + (1-\lambda)\boldsymbol{x}_1\rangle = 0.$$

We thus have

$$|\langle \mathbf{A}'_i, \boldsymbol{x}_1\rangle| = \lambda|\langle \mathbf{A}'_i, \boldsymbol{x} - \boldsymbol{x}_1\rangle| \leq \lambda\|\boldsymbol{x} - \boldsymbol{x}_1\|_2 \leq \delta.$$

Further we obtain that

$$\begin{aligned}\sup_{\boldsymbol{x}_1\in\mathcal{N}_\delta} \sup_{\substack{\boldsymbol{x}\in K\\ \|\boldsymbol{x}-\boldsymbol{x}_1\|_2\leq\delta}} d_{\mathbf{A}}(\boldsymbol{x}, \boldsymbol{x}_1) &= \sup_{\boldsymbol{x}_1\in\mathcal{N}_\delta} \sup_{\substack{\boldsymbol{x}\in K\\ \|\boldsymbol{x}-\boldsymbol{x}_1\|_2\leq\delta}} \frac{1}{m}\sum_{i=1}^m \mathbb{1}(\mathrm{sign}\left(\langle \mathbf{A}'_i, \boldsymbol{x}\rangle\right) \neq \mathrm{sign}\left(\langle \mathbf{A}'_i, \boldsymbol{x}_1\rangle\right))\\ &\leq \sup_{\boldsymbol{x}_1\in\mathcal{N}_\delta} \frac{1}{m}\sum_{i=1}^m \mathbb{1}\{|\langle \mathbf{A}_i, \boldsymbol{x}_1\rangle| \leq \delta\}.\end{aligned}$$

Combining the above result with (C.2), we obtain that, with probability at least $1 - \exp(-\delta^2 m)$,

$$\sup_{\boldsymbol{x}_1\in\mathcal{N}_\delta} \sup_{\substack{\boldsymbol{x}\in K\\ \|\boldsymbol{x}-\boldsymbol{x}_1\|_2\leq\delta}} d_{\mathbf{A}}(\boldsymbol{x}, \boldsymbol{x}_1) \leq (C+1)\delta. \tag{C.3}$$

For any points $\boldsymbol{x}, \boldsymbol{y} \in K$, let $\boldsymbol{x}_1, \boldsymbol{y}_1$ be their nearest points in $\mathcal{N}_\delta$. We have

$$
\begin{aligned}
&|d(\boldsymbol{x},\boldsymbol{y}) - d_{\mathbf{A}}(\boldsymbol{x},\boldsymbol{y})| \le |d(\boldsymbol{x},\boldsymbol{y}) - d(\boldsymbol{x}_1,\boldsymbol{y}_1)| + |d(\boldsymbol{x}_1,\boldsymbol{y}_1) - d_{\mathbf{A}}(\boldsymbol{x},\boldsymbol{y})| \\
&\overset{(a)}{\le} |d(\boldsymbol{x}_1,\boldsymbol{y}_1) - d_{\mathbf{A}}(\boldsymbol{x},\boldsymbol{y})| + 2\delta \le |d(\boldsymbol{x}_1,\boldsymbol{y}_1) - d_{\mathbf{A}}(\boldsymbol{x}_1,\boldsymbol{y}_1)| + |d_{\mathbf{A}}(\boldsymbol{x}_1,\boldsymbol{y}_1) - d_{\mathbf{A}}(\boldsymbol{x},\boldsymbol{y})| + 2\delta \\
&\overset{(b)}{\le} |d_{\mathbf{A}}(\boldsymbol{x}_1,\boldsymbol{y}_1) - d_{\mathbf{A}}(\boldsymbol{x},\boldsymbol{y})| + 3\delta \le |d_{\mathbf{A}}(\boldsymbol{x}_1,\boldsymbol{y}_1) - d_{\mathbf{A}}(\boldsymbol{x}_1,\boldsymbol{y})| + |d_{\mathbf{A}}(\boldsymbol{x}_1,\boldsymbol{y}) - d_{\mathbf{A}}(\boldsymbol{x},\boldsymbol{y})| + 3\delta \\
&\overset{(c)}{\le} d_{\mathbf{A}}(\boldsymbol{y}_1,\boldsymbol{y}) + d_{\mathbf{A}}(\boldsymbol{x}_1,\boldsymbol{x}) + 3\delta \overset{(d)}{\le} (2C+5)\delta,
\end{aligned}
$$

where (a) follows from

$$
|d(\boldsymbol{x},\boldsymbol{y}) - d(\boldsymbol{x}_1,\boldsymbol{y}_1)| \le |d(\boldsymbol{x},\boldsymbol{y}) - d(\boldsymbol{x}_1,\boldsymbol{y})| + |d(\boldsymbol{x}_1,\boldsymbol{y}) - d(\boldsymbol{x},\boldsymbol{y}_1)| \le d(\boldsymbol{x},\boldsymbol{x}_1) + d(\boldsymbol{x}_1,\boldsymbol{y}_1) \le 2\delta,
$$

step (b) follows from (C.1), step (c) follows from the triangle inequality of Hamming distance, step (d) is from (C.3). □

References

Nir Ailon and Bernard Chazelle. Approximate nearest neighbors and the fast johnson-lindenstrauss transform. In *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pages 557–563. ACM, 2006.

Nir Ailon and Edo Liberty. An almost optimal unrestricted fast johnson-lindenstrauss transform. *ACM Transactions on Algorithms (TALG)*, 9(3):21, 2013.

Nir Ailon and Holger Rauhut. Fast and rip-optimal transforms. *Discrete & Computational Geometry*, 52(4): 780–798, 2014.

Noga Alon. Problems and results in extremal combinatoricsâĂŤi. *Discrete Mathematics*, 273(1):31–53, 2003.

Alexandr Andoni and Piotr Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In *Foundations of Computer Science, 2006. FOCS'06. 47th Annual IEEE Symposium on*, pages 459–468. IEEE, 2006.

Mahdi Cheraghchi, Venkatesan Guruswami, and Ameya Velingker. Restricted isometry of fourier matrices and list decodability of random linear codes. *SIAM Journal on Computing*, 42(5):1888–1914, 2013.

Yunchao Gong and Svetlana Lazebnik. Iterative quantization: A procrustean approach to learning binary codes. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 817–824, 2011.

Yunchao Gong, Sanjiv Kumar, Henry A Rowley, and Svetlana Lazebnik. Learning binary codes for high-dimensional data using bilinear projections. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 484–491. IEEE, 2013.

Laurent Jacques, Jason N Laska, Petros T Boufounos, and Richard G Baraniuk. Robust 1-bit compressive sensing via binary stable embeddings of sparse vectors. *arXiv preprint arXiv:1104.3160*, 2011.

TS Jayram and David P Woodruff. Optimal bounds for johnson-lindenstrauss transforms and streaming problems with subconstant error. *ACM Transactions on Algorithms (TALG)*, 9(3):26, 2013.

William B Johnson and Joram Lindenstrauss. Extensions of lipschitz mappings into a hilbert space. *Contemporary mathematics*, 26(189-206):1, 1984.

Felix Krahmer and Rachel Ward. New and improved johnson-lindenstrauss embeddings via the restricted isometry property. *SIAM Journal on Mathematical Analysis*, 43(3):1269–1281, 2011.

Felix Krahmer, Shahar Mendelson, and Holger Rauhut. Suprema of chaos processes and the restricted isometry property. *Communications on Pure and Applied Mathematics*, 67(11):1877–1904, 2014.

Michel Ledoux and Michel Talagrand. *Probability in Banach Spaces: isoperimetry and processes*, volume 23. Springer, 1991.

Wei Liu, Jun Wang, Sanjiv Kumar, and Shih-Fu Chang. Hashing with graphs. In *Proceedings of the 28th International Conference on Machine Learning*, 2011.

Jelani Nelson, Eric Price, and Mary Wootters. New constructions of rip matrices with fast multiplication and fewer rows. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1515–1528. SIAM, 2014.

Mohammad Norouzi, David M Blei, and Ruslan Salakhutdinov. Hamming distance metric learning. In *Advances in Neural Information Processing Systems*, pages 1061–1069, 2012.

Yaniv Plan and Roman Vershynin. Robust 1-bit compressed sensing and sparse logistic regression: A convex programming approach. *Information Theory, IEEE Transactions on*, 59(1):482–494, 2013.

Yaniv Plan and Roman Vershynin. Dimension reduction by random hyperplane tessellations. *Discrete & Computational Geometry*, 51(2):438–461, 2014.

Maxim Raginsky and Svetlana Lazebnik. Locality-sensitive binary codes from shift-invariant kernels. In *Advances in neural information processing systems*, pages 1509–1517, 2009.

Mark Rudelson and Roman Vershynin. On sparse reconstruction from fourier and gaussian measurements. *Communications on Pure and Applied Mathematics*, 61(8):1025–1045, 2008.

Ruslan Salakhutdinov and Geoffrey Hinton. Semantic hashing. *International Journal of Approximate Reasoning*, 50(7):969–978, 2009.

Yair Weiss, Antonio Torralba, and Rob Fergus. Spectral hashing. In *Advances in neural information processing systems*, pages 1753–1760, 2009.

Felix X Yu, Sanjiv Kumar, Yunchao Gong, and Shih-Fu Chang. Circulant binary embedding. *arXiv preprint arXiv:1405.3162*, 2014.