# On Using Toeplitz and Circulant Matrices for Johnson-Lindenstrauss Transforms

Casper Benjamin Freksen[*] Kasper Green Larsen[*]

**Abstract**

The Johnson-Lindenstrauss lemma is one of the corner stone results in dimensionality reduction. It says that given $N$, for any set of $N$ vectors $X \subset \mathbb{R}^n$, there exists a mapping $f : X \to \mathbb{R}^m$ such that $f(X)$ preserves all pairwise distances between vectors in $X$ to within $(1 \pm \varepsilon)$ if $m = \mathcal{O}(\varepsilon^{-2} \lg N)$. Much effort has gone into developing fast embedding algorithms, with the Fast Johnson-Lindenstrauss transform of Ailon and Chazelle being one of the most well-known techniques. The current fastest algorithm that yields the optimal $m = \mathcal{O}(\varepsilon^{-2} \lg N)$ dimensions has an embedding time of $\mathcal{O}(n \lg n + \varepsilon^{-2} \lg^3 N)$. An exciting approach towards improving this, due to Hinrichs and Vybíral, is to use a random $m \times n$ Toeplitz matrix for the embedding. Using Fast Fourier Transform, the embedding of a vector can then be computed in $\mathcal{O}(n \lg m)$ time. The big question is of course whether $m = \mathcal{O}(\varepsilon^{-2} \lg N)$ dimensions suffice for this technique. If so, this would end a decades long quest to obtain faster and faster Johnson-Lindenstrauss transforms. The current best analysis of the embedding of Hinrichs and Vybíral shows that $m = \mathcal{O}(\varepsilon^{-2} \lg^2 N)$ dimensions suffices. The main result of this paper, is a proof that this analysis unfortunately cannot be tightened any further, i.e., there exists a set of $N$ vectors requiring $m = \Omega(\varepsilon^{-2} \lg^2 N)$ for the Toeplitz approach to work.

## 1 Introduction

The performance of many geometric algorithms depends heavily on the dimension of the input data. A widely used technique to combat this “curse of dimensionality”, is to preprocess the input via *dimensionality reduction* while approximately preserving important geometric properties. Running the algorithm on the lower dimensional data then uses less resources (time, space, etc.) and an approximate result for the high dimensional data can be derived from the low dimensional result.

Dimensionality reduction approximately preserving pairwise Euclidean distances has found uses in a wide variety of applications, including: Nearest-neighbour search [2, 12], clustering [6, 8], linear programming [22], streaming algorithms [19], compressed sensing [7, 11], numerical linear algebra [25], graph sparsification [20], and differential privacy [5]. See more applications in [21, 14]. The most fundamental result in this regime is the Johnson-Lindenstrauss (JL) lemma [15], which says the following:

**Theorem 1** (Johnson-Lindenstrauss lemma)**.** *Let $X \subset \mathbb{R}^n$ be a set of $N$ vectors, then for any $0 < \varepsilon < 1/2$, there exists a map $f : X \to \mathbb{R}^m$ for some $m = \mathcal{O}(\varepsilon^{-2} \lg N)$ such that*

$$\forall x, y \in X, (1 - \varepsilon)\|x - y\|_2^2 \le \|f(x) - f(y)\|_2^2 \le (1 + \varepsilon)\|x - y\|_2^2.$$

This result dates back to 1984 and says that to preserve pairwise Euclidean distances amongst a set of $N$ points/vectors in $\mathbb{R}^n$ to within a factor $(1 \pm \varepsilon)$, it suffices to use just $m = \mathcal{O}(\varepsilon^{-2} \lg N)$ dimensions. The bound on $m$ was very recently proven optimal [18].

The standard technique for constructing a map with the properties of Theorem 1 is the following: Let $A$ be an $m \times n$ matrix with entries independently sampled as either $\mathcal{N}(0, 1)$ random variables (as in [10]) or Rademacher (uniform among $\{-1, +1\}$) random variables (as in [1]). Once such entries have been drawn, let $f : \mathbb{R}^n \to \mathbb{R}^m$ be defined as:

$$f(x) = \frac{1}{\sqrt{m}} Ax.$$

[*] This research is supported by a Villum Young Investigator Grant, an AUFF Starting Grant and MADALGO, Center for Massive Data Algorithmics, a Center of the Danish National Research Foundation, grant DNRF84. Department of Computer Science, Aarhus University. {cfreksen,larsen}@cs.au.dk.

To prove that the map $f$ satisfies the guarantees in Theorem 1, it is first shown that for any vector $x$, the probability that $\|f(x)\|_2^2$ is not within $(1 \pm \varepsilon)\|x\|_2^2$ is less than $1/N^2$. This probability is called the error probability and denoted $\delta$. Using linearity of $f$ and a union bound over all pairs $x, y \in X$, the probability that all pairwise distances (i.e. the norm of the vector $x - y$) are preserved can be shown to be at least 1/2.

## 1.1 Time Complexity

Examining the classic Johnson-Lindenstrauss reduction above, we see that to embed a vector, we need to multiply with a dense matrix and the embedding time becomes $\mathcal{O}(nm)$ (or equivalently $\mathcal{O}(n\varepsilon^{-2} \lg N)$). This may be prohibitively large for many applications (recall one prime usage of dimensionality reduction is to speed up algorithms), and much research has been devoted to obtaining faster embedding time.

**Fast Johnson-Lindenstrauss Transform.** Ailon and Chazelle [2] were the first to address the question of faster Johnson-Lindenstrauss transforms. In their seminal paper, they introduced the so-called Fast Johnson-Lindenstrauss transform for speeding up dimensionality reduction. The basic idea in their paper is to first "precondition" the input data by multiplying with a diagonal matrix with random signs, followed by multiplying with a Hadamard matrix. This has the effect of "spreading" out the mass of the input vectors, allowing for the dense matrix $A$ above to be replaced with a sparse matrix. Since we can multiply with a Hadamard matrix using Fast Fourier Transform, this gives an embedding time of $\mathcal{O}(n \lg n + \varepsilon^{-2} \lg^3 N)$ for embedding into the optimal $m = \mathcal{O}(\varepsilon^{-2} \lg N)$ dimensions. For $m = \varepsilon^{-2} \lg N \leq n^{1/2-\gamma}$ for any constant $\gamma > 0$, the embedding complexity was improved even further down to $\mathcal{O}(n \lg m)$ in [3].

Another approach to achieve the $\mathcal{O}(n \lg m)$ embedding time, but without the restriction on $\varepsilon^{-2} \lg N \leq n^{1/2-\gamma}$, is to sacrifice the target dimension. This was done in [4] and later improved in [17], where the embedding complexity was $\mathcal{O}(n \lg m)$ at the cost of an increased target dimension $m = \mathcal{O}(\varepsilon^{-2} \lg N \lg^4 n)$.

**Sparse Vectors.** Another approach to improve the performance of JL transforms, is to assume the input data is sparse, i.e. has few non-zero coordinates. Designing an algorithm based on the work in [24], Dasgupta et al. [9] achieved an embedding complexity of $\mathcal{O}(\|x\|_0 \varepsilon^{-1} \lg^2(mN) \lg N)$, where $\|x\|_0 = |\{i \mid x_i \neq 0\}|$. This was later improved to $\mathcal{O}(\|x\|_0 \varepsilon^{-1} \lg N)$ in [16].

**Toeplitz Matrices.** Finally, another very exciting approach is to use Toeplitz matrices or partial circulant matrices for the embedding. We first introduce the terminology.

An $m \times n$ Toeplitz matrix is an $m \times n$ matrix, where every entry on a diagonal has the same value:

$$\begin{pmatrix} t_0 & t_1 & t_2 & \cdots & t_{n-1} \\ t_{-1} & t_0 & t_1 & \cdots & t_{n-2} \\ t_{-2} & t_{-1} & t_0 & \cdots & t_{n-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ t_{-(m-1)} & t_{-(m-2)} & t_{-(m-3)} & \cdots & t_{n-m} \end{pmatrix}$$

A partial circulant matrix is a special kind of Toeplitz matrix, where every row, except the first, is the previous row rotated once:

$$\begin{pmatrix} t_0 & t_1 & t_2 & \cdots & t_{n-1} \\ t_{n-1} & t_0 & t_1 & \cdots & t_{n-2} \\ t_{n-2} & t_{n-1} & t_0 & \cdots & t_{n-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ t_{n-(m-1)} & t_{n-(m-2)} & t_{n-(m-3)} & \cdots & t_{n-m} \end{pmatrix}$$

Hinrichs and Vybíral [13] proposed the following algorithm for generating a JL embedding based on a Toeplitz matrix[1]: Let $t_{-(m-1)}, t_{-(m-2)}, \ldots, t_{n-1}$ and $d_1, \ldots, d_n$ be i.i.d. Rademacher random variables, and $T$ be a Toeplitz matrix defined from $t_{-(m-1)}, t_{-(m-2)}, \ldots, t_{n-1}$ such that entry $(i, j)$ takes values

[1] [13] uses a partial circulant matrix but notes that a Toeplitz matrix could be used as well.

Table 1: Comparison of the performances of various Johnson-Lindenstrauss transform algorithms. $N$ is the number of input vectors, $n$ is the dimension of the input vectors, $m$ is the dimension of the output vectors, $\varepsilon$ is the distortion.

| Type | Embedding time | Target dimension ($m$) | Ref. | Notes |
|---|---|---|---|---|
| Random projection | $\mathcal{O}(nm)$ | $\mathcal{O}(\varepsilon^{-2} \lg N)$ | [10] | |
| Sparse | $\mathcal{O}(\|x\|_0 \varepsilon^{-1} \lg^2(mN) \lg N)$ | $\mathcal{O}(\varepsilon^{-2} \lg N)$ | [9] | |
| Sparse | $\mathcal{O}(\|x\|_0 \varepsilon^{-1} \lg N)$ | $\mathcal{O}(\varepsilon^{-2} \lg N)$ | [16] | |
| FFT | $\mathcal{O}(n \lg n + m \lg^2 N)$ | $\mathcal{O}(\varepsilon^{-2} \lg N)$ | [2] | |
| FFT | $\mathcal{O}(n \lg m)$ | $\mathcal{O}(\varepsilon^{-2} \lg N)$ | [3] | $m \le n^{1/2-\gamma}$ |
| FFT | $\mathcal{O}(n \lg m)$ | $\mathcal{O}(\varepsilon^{-2} \lg N \lg^4 n)$ | [17] | |
| Toeplitz | $\mathcal{O}(n \lg m)$ | $\mathcal{O}(\varepsilon^{-2} \lg^3 N)$ | [13] | |
| Toeplitz | $\mathcal{O}(n \lg m)$ | $\mathcal{O}(\varepsilon^{-2} \lg^2 N)$ | [23] | |

$t_{j-i}$ for $i = 1, \dots, m$ and $j = 1, \dots, n$. Let $D$ be an $n \times n$ diagonal matrix with the random variable $d_i$ giving the $i$'th diagonal entry. Define the map $f$ as

$$f(x) = \frac{1}{\sqrt{m}} TDx.$$

Multiplying with a Toeplitz matrix corresponds to computing a convolution and can be done using Fast Fourier Transform. By appropriately blocking the input coordinates, the complexity of embedding a vector $x$ is just $\mathcal{O}(n \lg m)$ for any target dimension $m$. The big question is of course, how low can the target dimension $m$ be, while preserving the distances between vectors up to a factor of $1 \pm \varepsilon$?

In the original paper [13], the authors proved that setting the target dimension to $m = \mathcal{O}(\varepsilon^{-2} \lg^3(1/\delta))$, the norm of any vector would be preserved to within $(1 \pm \varepsilon)$ with probability at least $1 - \delta$. Setting $\delta = 1/N^2$, a union bound over all pairwise difference vectors (as in the classic construction) shows that dimension $m = \mathcal{O}(\varepsilon^{-2} \lg^3 N)$ suffices. Later, the analysis was refined in [23], which lowered the target dimension to $m = \mathcal{O}(\varepsilon^{-2} \lg^2(1/\delta))$ for preserving norms to within $(1 \pm \varepsilon)$ with probability $1 - \delta$. Again, setting $\delta = 1/N^2$, this gives $m = \mathcal{O}(\varepsilon^{-2} \lg^2 N)$ target dimension. Now if the analysis could be tightened even further to give the optimal $m = \mathcal{O}(\varepsilon^{-2} \lg N)$ dimensions, this would end the decades long quest for faster and faster embedding algorithms!

**Our Contribution.** Our main result unfortunately shows that the analysis of Vybíral [23] cannot be tightened to give an even lower target dimensionality. More specifically, we prove that the upper bound given in [23] is optimal:

**Theorem 2.** *Let $T$ and $D$ be the $m \times n$ Toeplitz and $n \times n$ diagonal matrix in the embedding proposed by [13]. For all $0 < \varepsilon < C$, where $C$ is a universal constant, and any desired error probability $\delta > 0$, if the following holds for every unit vector $x \in \mathbb{R}^n$:*

$$\Pr\left[\left|\left\|\frac{1}{\sqrt{m}} TDx\right\|_2^2 - 1\right| < \varepsilon\right] > 1 - \delta,$$

*then it must be the case that $m = \Omega(\varepsilon^{-2} \lg^2(1/\delta))$.*

While Theorem 2 already shows that one cannot tighten the analysis of Vybíral for preserving the norm of just one vector, Theorem 2 does leave open the possibility that one would not need to union bound over all $N^2$ pairs of difference vectors when trying to preserve all pairwise distances amongst a set of $N$ vectors. It could still be the case that there somehow was a strong positive correlation between distances being preserved (though this seems extremely unlikely, and would be something not seen in any previous approach to JL). To complete the picture, we indeed show that this is not the case, at least for $N$ somewhat smaller than the dimension $n$:

**Theorem 3.** *Let $T$ and $D$ be the $m \times n$ Toeplitz and $n \times n$ diagonal matrix in the embedding proposed by [13]. For all $0 < \varepsilon < C$, where $C$ is a universal constant, if the following holds for every set of $N$*

*vectors $X \subset \mathbb{R}^n$:*

$$\Pr\left[\forall x, y \in X : \left| \left\| \frac{1}{\sqrt{m}} TDx - \frac{1}{\sqrt{m}} TDy \right\|_2^2 - \|x - y\|_2^2 \right| \leq \varepsilon \|x - y\|_2^2 \right] = \Omega(1),$$

*then it must be the case that either $m = \Omega(\varepsilon^{-2} \lg^2 N)$ or $m = \Omega(n/N)$.*

We remark that our proofs also work if we replace $T$ be a partial circulant matrix (which was also proposed in [13]). Furthermore, we expect that minor technical manipulations to our proof would also show the above theorems when the entries of $T$ and $D$ are $\mathcal{N}(0, 1)$ distributed rather than Rademacher (this was also proposed in [13]).

## 2 Lower Bound for One Vector

Let $T$ be $m \times n$ Toeplitz matrix defined from random variables $t_{-(m-1)}, t_{-(m-2)}, \ldots, t_{n-1}$ such that entry $(i, j)$ takes values $t_{j-i}$ for $i = 1, \ldots, m$ and $j = 1, \ldots, n$. Let $D$ be an $n \times n$ diagonal matrix with the random variable $d_i$ giving the $i$'th diagonal entry. This section shows the following:

**Theorem 4.** *Let $T$ be $m \times n$ Toeplitz and $D$ $n \times n$ diagonal. If $t_{-(m-1)}, t_{-(m-2)}, \ldots, t_{n-1}$ and $d_1, \ldots, d_n$ are independently distributed Rademacher random variables for $i = -(m-1), \ldots, n-1$ and $j = 1, \ldots, n$, then for all $0 < \varepsilon < C$, where $C$ is a universal constant, there exists a unit vector $x \in \mathbb{R}^n$ such that*

$$\Pr\left[\left| \left\| \frac{1}{\sqrt{m}} TDx \right\|_2^2 - 1 \right| > \varepsilon \right] \geq 2^{-\mathcal{O}(\varepsilon\sqrt{m})}.$$

*and furthermore, all but the first $O(\sqrt{m})$ coordinates of $x$ are $0$.*

It follows from Theorem 4 that if we want to have probability at least $1 - \delta$ of preserving the norm of *any* unit vector $x$ to within $(1 \pm \varepsilon)$, it must be the case that $\varepsilon\sqrt{m} = \Omega(\lg(1/\delta))$, i.e. $m = \Omega(\varepsilon^{-2} \lg^2(1/\delta))$. This is precisely the statement of Theorem 2. Thus we set out to prove Theorem 4.

To prove Theorem 4, we wish to invoke the Paley-Zygmund inequality, which states, that if $X$ is a non-negative random variable with finite variance and $0 \leq \theta \leq 1$, then

$$\Pr[X > \theta\mathbb{E}[X]] \geq (1 - \theta)^2 \frac{\mathbb{E}^2[X]}{\mathbb{E}[X^2]}.$$

We carefully choose a unit vector $x$, and define the random variable for Paley-Zygmund to be the $k$'th moment of the difference between the norm of $x$ transformed and 1.

*Proof.* Let $k$ be an even positive integer less than $m/4$ and define $s := 4k$. Note that $s \leq m$. Let $x$ be an arbitrary $n$-dimensional unit vector such that the first $s$ coordinates are in $\{-1/\sqrt{s}, +1/\sqrt{s}\}$, while the remaining $n - s$ coordinates are 0. Define the random variable parameterized by $k$

$$Z_k := \left( \left\| \frac{1}{\sqrt{m}} TDx \right\|_2^2 - 1 \right)^k.$$

Since $k$ is even, the random variable $Z_k$ is non-negative.

We wish to lower bound $\mathbb{E}[Z_k]$ and upper bound $\mathbb{E}[Z_k^2]$ in order to invoke Paley-Zygmund. The bounds we prove are as follows:

**Lemma 5.** *If $k \leq \sqrt{m}$, then the random variable $Z_k$ satisfies:*

$$\mathbb{E}[Z_k] \geq m^{-k/2} k^k 2^{-\mathcal{O}(k)}$$

*and*

$$\mathbb{E}[Z_k^2] \leq m^{-k} k^{2k} 2^{\mathcal{O}(k)}.$$

Before proving Lemma 5 we show how to use it together with Paley-Zygmund to complete the proof of Theorem 4.

We start by invoking Paley-Zygmund and then rewriting the expectations according to Lemma 5,

$$\begin{aligned}
\Pr[Z_k > \mathbb{E}[Z_k]/2] \geq (1/4)\frac{\mathbb{E}^2[Z_k]}{\mathbb{E}[Z_k^2]} &\implies \\
\Pr[Z_k^{1/k} > (\mathbb{E}[Z_k]/2)^{1/k}] \geq (1/4)\frac{\mathbb{E}^2[Z_k]}{\mathbb{E}[Z_k^2]} &\implies \\
\Pr\left[\left|\left\|\frac{1}{\sqrt{m}}TDx\right\|_2^2 - 1\right| > \frac{k}{C_0\sqrt{m}}\right] \geq 2^{-\mathcal{O}(k)}. &
\end{aligned}$$

Here $C_0$ is some constant greater than 0. For any $0 < \varepsilon < 1/C_0$, we can now set $k$ such that $k/(C_0\sqrt{m}) = \varepsilon$, i.e. we choose $k = \varepsilon C_0\sqrt{m}$. This choice of $k$ satisfies $k \leq \sqrt{m}$ as required by Lemma 5. We have thus shown that:

$$\Pr\left[\left|\left\|\frac{1}{\sqrt{m}}TDx\right\|_2^2 - 1\right| > \varepsilon\right] \geq 2^{-\mathcal{O}(\varepsilon\sqrt{m})}.$$

□

*Remark.* Theorem 4 can easily be extended to partial circulant matrices. The difference between partial circulant and Toeplitz matrices is the dependence between the values in the first $m$ and last $m$ columns. However, as only the first $s = 4k \leq 4\sqrt{m}$ entries in $x$ are nonzero, the last $m$ columns are ignored, and so partial circulant and Toeplitz matrices behave identically in our proof.

*Proof of Lemma 5.* Before we prove the two bounds in Lemma 5 individually, we rewrite $\mathbb{E}[Z_k]$, as this benefits both proofs.

$$\begin{aligned}
\mathbb{E}[Z_k] &= \mathbb{E}\left[\left(\left\|\frac{1}{\sqrt{m}}TDx\right\|_2^2 - 1\right)^k\right] \\
&= \mathbb{E}\left[\left(\left(\frac{1}{m}\sum_{i=1}^m\left(\sum_{j=1}^n t_{j-i}d_jx_j\right)^2\right) - 1\right)^k\right] \\
&= \mathbb{E}\left[\left(\left(\frac{1}{m}\sum_{i=1}^m\left(\left(\sum_{j=1}^n t_{j-i}^2d_j^2x_j^2\right) + \left(\sum_{j=1}^n\sum_{h\in\{1,\dots,n\}\setminus\{j\}} t_{j-i}t_{h-i}d_jd_hx_jx_h\right)\right)\right) - 1\right)^k\right] \\
&= \mathbb{E}\left[\left(\frac{1}{m}\sum_{i=1}^m\left(\left(\sum_{j=1}^n t_{j-i}^2d_j^2x_j^2 - x_j^2\right) + \left(\sum_{j=1}^n\sum_{h\in\{1,\dots,n\}\setminus\{j\}} t_{j-i}t_{h-i}d_jd_hx_jx_h\right)\right)\right)^k\right] \\
&= \mathbb{E}\left[\left(\frac{1}{m}\sum_{i=1}^m\sum_{j=1}^n\sum_{h\in\{1,\dots,n\}\setminus\{j\}} t_{j-i}t_{h-i}d_jd_hx_jx_h\right)^k\right] \\
&= \frac{1}{m^k}\sum_{S\in([m]\times[n]\times[n])^k|\forall(i,j,h)\in S:h\neq j}\mathbb{E}\left[\prod_{(i,j,h)\in S} t_{j-i}t_{h-i}d_jd_hx_jx_h\right]
\end{aligned}$$

Observe that for $j > s$ or $h > s$ the product becomes 0, as either $x_j$ or $x_h$ is 0. By removing all these terms, we simplify the sum to

$$\mathbb{E}[Z_k] = \frac{1}{m^k}\sum_{S\in([m]\times[s]\times[s])^k|\forall(i,j,h)\in S:h\neq j}\mathbb{E}\left[\prod_{(i,j,h)\in S} t_{j-i}t_{h-i}d_jd_hx_jx_h\right]$$

Observe for an $S \in ([m] \times [s] \times [s])^k$, that the value $\mathbb{E}\left[\prod_{(i,j,h)\in S} t_{j-i} t_{h-i} d_j d_h x_j x_h\right]$ is 0 if one of the following two things are true:

- A $d_j$ occurs an odd number of times in the product.
- A variable $t_a$ occurs an odd number of times in the product.

To see this, note that by the independence of the random variables, we can write the expectation of the product, as a product of expectations where each term in the product has all the occurrences of the same random variable. Since the $d_j$'s and $t_a$'s are Rademachers, the expectation of any odd power of one of these random variables is 0. Thus if just a single random variable amongst the $d_j$'s and $t_a$'s occurs an odd number of times, we have $\mathbb{E}\left[\prod_{(i,j,h)\in S} t_{j-i} t_{h-i} d_j d_h x_j x_h\right] = 0$. Similarly, we observe that if every random variable occurs an even number of times, then the expectation of the product is exactly $1/s^k$ since each $x_j$ also occurs an even number of times. If $\Gamma_k$ denotes the number of tuples $S \in ([m] \times [s] \times [s])^k$ such that $\forall (i,j,h) \in S$ we have $h \neq j$ and furthermore:

- For all columns $a \in [s]$, $|\{(i,j,h) \in S \mid j = a \vee h = a\}| \mod 2 = 0$.
- For all diagonals $a \in \{-(m-1), \ldots, s-1\}$, $|\{(i,j,h) \in S \mid j - i = a \vee h - i = a\}| \mod 2 = 0$.

Then we conclude

$$\mathbb{E}[Z_k] = \frac{\Gamma_k}{s^k m^k}. \tag{1}$$

Note that $Z_k^2 = Z_{2k}$. Therefore,

$$\mathbb{E}[Z_k^2] = \mathbb{E}[Z_{2k}] = \frac{\Gamma_{2k}}{s^{2k} m^{2k}}. \tag{2}$$

To complete the proof of Lemma 5 we need lower and upper bounds for $\Gamma_k$ and $\Gamma_{2k}$. The bounds we prove are

**Lemma 6.** *If $k \leq \sqrt{m}$, then $\Gamma_k$ and $\Gamma_{2k}$ satisfy:*

$$\Gamma_k = m^{k/2} s^k k^k 2^{-\mathcal{O}(k)}$$

*and*

$$\Gamma_{2k} = m^k s^{2k} k^{2k} 2^{\mathcal{O}(k)}.$$

The proofs of the two bounds in Lemma 6 are in Sections 2.1 and 2.2.

Substituting the bounds from Lemma 6 in (1) and (2) we get

$$\mathbb{E}[Z_k] = m^{-k/2} k^k 2^{-\mathcal{O}(k)}$$
$$\mathbb{E}[Z_k^2] = m^{-k} k^{2k} 2^{-\mathcal{O}(k)},$$

which are the bounds we sought for Lemma 5. □

## 2.1 Lower Bounding $\Gamma_k$

We first recall that the definition of $\Gamma_k$ is the number of tuples $S \in ([m] \times [s] \times [s])^k$ satisfying that $\forall (i,j,h) \in S$ we have $h \neq j$ and furthermore:

- For all columns $a \in [s]$, $|\{(i,j,h) \in S \mid j = a \vee h = a\}| \mod 2 = 0$.
- For all diagonals $a \in \{-(m-1), \ldots, s-1\}$, $|\{(i,j,h) \in S \mid j - i = a \vee h - i = a\}| \mod 2 = 0$.

We view a triple $(i,j,h) \in ([m] \times [s] \times [s])$ as two entries $(i,j)$ and $(i,h)$ in an $m \times s$ matrix. Furthermore, when we say that a triple *touches* a column or diagonal, a matrix entry of the triple lie on that column or diagonal, so $(i,j,h)$ touches columns $j$ and $h$ and diagonals $j-i$ and $h-i$. Similarly, we say that a tuple $S \in ([m] \times [s] \times [s])^k$ touches a given column or diagonal $l$ times, if $l$ triples in $S$ touches that column or diagonal.

We intent to prove a lower bound for $\Gamma_k$ by constructing a big family of tuples $\mathcal{F} \subseteq ([m] \times [s] \times [s])^k$, where each tuple satisfies, that each column and diagonal touched by that tuple is touched exactly twice. As each column and diagonal is touched an even number of times, the number of tuples in the family is a lower bound for $\Gamma_k$.

*Proof of* $\Gamma_k = m^{k/2}s^k k^k 2^{-\mathcal{O}(k)}$. We describe how to construct a family of tuples $\mathcal{F} \subseteq ([m] \times [s] \times [s])^k$ satisfying that $\forall S \in \mathcal{F}, \forall (i,j,h) \in S$ we have $h \neq j$ and furthermore:

- For all columns $a \in [s]$, $|\{(i,j,h) \in S \mid j = a \vee h = a\}| \in \{0,2\}$.
- For all diagonals $a \in \{-(m-1),\dots,s-1\}$, $|\{(i,j,h) \in S \mid j-i = a \vee h-i = a\}| \in \{0,2\}$.

From this and the definition of $\Gamma_k$ it is clear that $|\mathcal{F}| \leq \Gamma_k$.

When constructing an $S \in \mathcal{F}$, we view $S$ as consisting of two halves $S_1$ and $S_2$, such that $S_1$ touches exactly the same columns and diagonals as $S_2$ and both $S_1$ and $S_2$ touches each column and diagonal at most once. To capture this, we give the following definition, where $\mathbb{S}$ is meant to be the family of such halves $S_1$ and $S_2$.

**Definition 7.** Let $\mathbb{S}$ be the set of all tuples $S \in ([m] \times [s] \times [s])^{k/2}$ such that

- $\forall (i,j,h) \in S, j \neq h$
- For all columns $a \in [s], |\{(i,j,h) \in S \mid j = a \vee h = a\}| \leq 1$
- For all diagonals $a \in \{-(m-1),\dots,s-1\}, |\{(i,j,h) \in S \mid j-i = a \vee h-i = a\}| \leq 1$

Definition 7 mimics the definition of $\Gamma_k$, and the first item in Definition 7 ensures that the triples in a tuple in $\mathbb{S}$ are of the same form as in $\Gamma_k$. The final two items ensure that each column and diagonal, respectively, is touched at most once. This is exactly the properties we wanted of $S_1$ and $S_2$ individually.

We can now construct $\mathcal{F}$ as all pairs of (half) tuples $S_1, S_2 \in \mathbb{S}$, such that $S_1$ touches exactly the same columns and diagonals as $S_2$. To capture that $S_1$ and $S_2$ touch the same columns and diagonals, we introduce the notion of a signature. A signature of $S_i$ is the set of columns and diagonals touched by $S_i$.

To have $S_1$ and $S_2$ touch exactly the same columns and diagonals, it is necessary and sufficient that they have the same signature.

We introduce the following notation: $B$ denotes the number of signatures with at least one member, and by enumerating the signatures, we let $b_i$ denote the number of (half) tuples in $\mathbb{S}$ with signature $i$.

We recall that a (half) tuple $S_1 \in \mathbb{S}$ touches each column and diagonal at most once, and if $S_1$ and $S_2$ share the same signature, they touch exactly the same columns and diagonals. Therefore, using $\circ$ to mean concatenation, $S = S_1 \circ S_2 \in \mathcal{F}$, as each column and diagonal touched is touched exactly twice. Therefore $|\mathcal{F}|$ is a lower bound for $\Gamma_k$. Note that for a given signature $i$, the number of choices of $S_1$ and $S_2$ with that signature is $b_i^2$. This gives the following inequality,

$$\Gamma_k \geq |\mathcal{F}| = \sum_{i=1}^{B} b_i^2.$$

We now apply the Cauchy-Schwarz inequality:

$$\sum_{i=1}^{B} b_i^2 \sum_{i=1}^{B} 1^2 \geq \Big(\sum_{i=1}^{B} b_i\Big)^2 \implies \sum_{i=1}^{B} b_i^2 \geq \frac{\left(\sum_{i=1}^{B} b_i\right)^2}{\sum_{i=1}^{B} 1^2} \implies \Gamma_k \geq \frac{|\mathbb{S}|^2}{B}. \tag{3}$$

To get a lower bound on $|\mathbb{S}|^2/B$ (and in turn $\Gamma_k$), we need a lower bound on $|\mathbb{S}|$ and an upper bound on $B$. These bounds are stated in the following lemmas

**Lemma 8.** $|\mathbb{S}| = \Omega(m^{k/2}s^k 2^{-k})$.

**Lemma 9.** $B = \mathcal{O}\left(\binom{m+s}{k/2} s^{k/2} \binom{s}{k}\right)$

Before proving any of these lemmas, we show that they together with (3) give the desired lower bound on $\Gamma_k$:

$$\Gamma_k = \frac{\Omega(m^{k/2}s^k 2^{-k})^2}{\mathcal{O}\left(\binom{m+s}{k/2} s^{k/2} \binom{s}{k}\right)} = \Omega\left(\frac{m^k s^{2k} 2^{-2k} (k/2)^{k/2} k^k}{(m+s)^{k/2} s^{k/2} s^k}\right). \tag{4}$$

Because $s = 4k$, we have $\frac{(k/2)^{k/2}}{s^{k/2}} = 2^{-\Theta(k)}$, and because $s \leq m$: $\frac{m^k}{(m+s)^{(k/2)}} = m^{k/2}2^{-\Theta(k)}$. With this we can simplify (4),

$$\Gamma_k = m^{k/2} s^k k^k 2^{-\mathcal{O}(k)}.$$

which is the lower bound we sought. □

*Proof of Lemma 8.* Recall that $\mathbb{S} \subseteq ([m] \times [s] \times [s])^{k/2}$ is the set of (half) tuples that touch each column and diagonal at most once, and, for each triple $(i, j, h)$ in these (half) tuples, we have $j \neq h$.

We prove Lemma 8 by analysing how we can create a large number of distinct $S \in \mathbb{S}$ by choosing the triples in $S$ iteratively.

For each triple, we choose a row and two distinct entries on this row. We choose the row among any of the $m$ rows.

However, because $S \in \mathbb{S}$, when choosing entries on the row, we cannot choose entries that lie on columns or diagonals touched by previously chosen triples. Instead we choose the two entries among any of the other entries. Therefore, whenever we choose a triple, this triple prevents at most four row entries from being chosen for every subsequent triple, as the two diagonals and two columns touched by the chosen triple intersect with at most four entries on the rows of the subsequent triples. This leads to the following recurrence, describing a lower bound for the number of triples

$$F(r, c, t) = \begin{cases} r \cdot c \cdot (c-1) \cdot F(r, c-4, t-1) & \text{if } t > 0 \\ 1 & \text{otherwise} \end{cases} \tag{5}$$

where $r$ is the number of rows to choose from, $c$ is the minimum number of choosable entries in any row, and $t$ is the number of triples left to choose.

Inspecting (5), we can see that $F$ can equivalently be defined as

$$F(r, c, t) = r^t \prod_{i=0}^{t-1} (c - 4i)(c - 1 - 4i). \tag{6}$$

If $t \leq \frac{c}{8}$ then the terms inside the product in (6) are greater than $\frac{c}{2}$, so we can bound $F$ from below:

$$F(r, c, t) \geq r^t \big(\frac{c}{2}\big)^{2t} = r^t c^{2t} \frac{1}{4^t}.$$

We now insert the values for $r, c$ and $t$ to find a lower bound for $|\mathbb{S}|$, noting that $s = 4k$ ensures that $t \leq \frac{c}{8}$:

$$|\mathbb{S}| \geq F\big(m, s, \frac{k}{2}\big) \geq m^{k/2} s^k \frac{1}{4^{k/2}} \implies |\mathbb{S}| = \Omega(m^{k/2} s^k 2^{-k}).$$

□

*Proof of Lemma 9.* Recall that for at triple $S \in \mathbb{S}$ we define the signature as the set of columns and diagonals touched by $S$. Furthermore, viewing a triple $(i, j, h) \in ([m] \times [s] \times [s])$ as the two entries $(i, j)$ and $(i, h)$ in an $m \times s$ matrix, we define the left endpoint as $(i, \min\{j, h\}$ and the right endpoint as $(i, \max\{j, h\})$.

The claim to prove is

$$B = \mathcal{O}\left( \binom{m+s}{k/2} s^{k/2} \binom{s}{k} \right).$$

This is proven by first showing an upper bound on the number of choices for the diagonals of left endpoints, then diagonals of right endpoints and finally for columns.

In an $m \times s$ matrix there are $m + s$ different diagonals and as the chosen diagonals have to be distinct, there are $\binom{m+s}{k/2}$ choices for the diagonals corresponding to left endpoints in a triple.

As the right endpoint of a triple has to be in the same row as the left endpoint, there are at most $s$ choices for the diagonal corresponding to the right endpoint when the left endpoint has been chosen (which it has in our case). This gives a total of $s^{k/2}$ choices for diagonals corresponding to right endpoints.

Finally, there are $s$ columns to choose from and the chosen columns have to be distinct, and so the total number of choices of columns is $\binom{s}{k}$.

The product of these number of choices gives the upper bound sought. □

## 2.2 Upper Bounding $\Gamma_{2k}$

*Proof.* Recall that $\Gamma_{2k}$ is defined as the number of tuples $S \in ([m] \times [s] \times [s])^{2k}$ such that $\forall (i, j, h) \in S$ we have $h \neq j$ and furthermore:

- For all columns $a \in [s]$, $|\{(i, j, h) \in S \mid j = a \vee h = a\}| \mod 2 = 0$.
- For all diagonals $a \in \{-(m-1), \dots, s-1\}$, $|\{(i, j, h) \in S \mid j - i = a \vee h - i = a\}| \mod 2 = 0$.

Let $\mathcal{F} \subseteq ([m] \times [s] \times [s])^{2k}$ be the family of tuples satisfying these conditions, and so $|\mathcal{F}| = \Gamma_{2k}$.

To prove an upper bound on $\Gamma_{2k}$, we show how to encode a tuple $S \in \mathcal{F}$ using at most $k \lg m + 2k \lg s + 2k \lg k + \mathcal{O}(k)$ bits, such that $S$ can be decoded from this encoding. Since any $S \in \mathcal{F}$ can be encoded using $k \lg m + 2k \lg s + 2k \lg k + \mathcal{O}(k)$ bits and $|\mathcal{F}| = \Gamma_{2k}$, we can conclude:

$$\Gamma_{2k} = 2^{k \lg m + 2k \lg s + 2k \lg k + \mathcal{O}(k)} = m^k s^{2k} k^{2k} 2^{\mathcal{O}(k)}.$$

Let $\sigma$ denote the encoding function and $\sigma^{-1}$ denote the decoding function. If $S \in \mathcal{F}$ and $t \in S$, $\sigma(t)$ denotes the encoding of the triple $t$, $\sigma(S)$ denotes the encoding of the entire tuple $S$, and $\sigma(\mathcal{F})$ denotes the image of $\sigma$.

A tuple $S \in \mathcal{F}$ consists of triples $t_1, t_2, \dots, t_{2k}$ such that $S = t_1 \circ t_2 \circ \cdots \circ t_{2k}$. To encode $S \in \mathcal{F}$ we encode each of the triples and store them in the same order: $\sigma(S) = \sigma(t_1) \circ \sigma(t_2) \circ \cdots \circ \sigma(t_{2k})$.

We will first describe a graph view of a tuple $S$ which will be useful for encoding and decoding, then we will show an encoding algorithm and finally a decoding algorithm.

**Graph** A tuple $S \in \mathcal{F}$ forms a (multi)graph structure, where every triple $(i, j, h) \in S$ is a vertex. Since $S \in \mathcal{F}$, there lies an even number of triple endpoints on each diagonal. We can thus pair endpoints lying on the same diagonal, such that every endpoint is paired with exactly one other endpoint. When two triples have endpoints that are paired, the triples have an edge between them in the graph. As every triple has two endpoints, every vertex has degree two, and so the graph consists entirely of simple cycles of length at least two.

**Encoding** To encode an $S \in \mathcal{F}$, we first encode each cycle by itself by defining the $\sigma(t)$'s for the triples $t$ in the cycle. After this, we order the defined $\sigma(t)$'s as the $t$'s were ordered in the input.

1. For each cycle we perform the following:

   (a) We pick any vertex $t = (i, j, h)$ of the cycle and give it the type head. Define $\sigma(t)$ as the concatenation of its type head, its row $i$, and two columns $j$ and $h$. This uses $\lg m + 2 \lg s + \mathcal{O}(1)$ bits.

   (b) We iterate through the cycle starting after head and give vertices, except the last, type mid. The last vertex just before head is given the type last.

   (c) For each triple $t$ of type mid we store its type and two columns explicitly. However, instead of storing its row we store the index of its predecessor in the cycle order as well as how they are connected: If we typed $t_r$ just before $t_s$ when iterating through the cycle, when encoding $t_s$ we store $r$ as well as whether $t_r$ and $t_s$ are connected by the left or right endpoint in $t_r$ and left or right endpoint in $t_s$. So define $\sigma(t)$ as the concatenation of mid, the two columns, the predecessor index and how it is connected to the predecessor. All in all we spend $\lg k + 2 \lg s + \mathcal{O}(1)$ bits encoding each mid.

   (d) Finally, to encode the triple $t$, which is typed last, we define $\sigma(t)$ as the concatenation of its type, its predecessors index, how it is connected to its predecessor, and the column of the endpoint on the predecessor's diagonal. We thus spend $\lg k + \lg s + \mathcal{O}(1)$ bits to encode a last. However, since $s = 4k$ the number of bits per encoded last is equivalent to $2 \lg k + \mathcal{O}(1)$, which turns out to simplify the analysis later.

   Note that for each triple, the type is encoded in the first 2 bits[2] in the encoding of the triple. This will be important during decoding.

[2] To encode the type in 2 bits we could use the following scheme: 00 = head, 01 = mid, 10 = last, and 11 is unused.

2. After encoding all cycles, we order the encoded triples in the same order as the triples in the input, and output the concatenation of the encoded triples: $\sigma(t_1) \circ \sigma(t_2) \circ \cdots \circ \sigma(t_{2k})$.

To analyse the number of bits needed in total, we look at the average number of bits per triple inside a cycle. Since all cycles have a length of at least two, each cycle has a $\mathsf{head}$ and a $\mathsf{last}$ triple. These two use an average of $\frac{\lg m}{2} + \lg s + \lg k + \mathcal{O}(1)$ bits. We now claim that the number of bits per $\mathsf{mid}$ is bounded by this average. Recall that we assumed $\sqrt{m} \geq k$ and $s = 4k$:

$$\sqrt{m} \geq k \implies \frac{\lg m}{2} \geq \lg k \implies \frac{\lg m}{2} + \lg s + \lg k + 2 \geq \lg k + 2 \lg s \implies$$
$$\frac{\lg m}{2} + \lg s + \lg k + \mathcal{O}(1) \geq |\mathsf{mid}|.$$

Since the average of all $2k$ triples is at most $\frac{\lg m}{2} + \lg s + \lg k + \mathcal{O}(1)$ bits, the number of bits for all triples is at most $k \lg m + 2k \lg s + 2k \lg k + \mathcal{O}(k)$.

**Decoding** We now show how to decode any $\sigma(S) \in \sigma(\mathcal{F})$ such that $\sigma^{-1}(\sigma(S)) = S$.

First we need to extract the encodings of the individual triples, then we decode each cycle, and finally we restore the order of triples. The following steps describes this algorithm in more detail.

1. We first extract the individual $\sigma(t_i)$'s, by iterating through the bit string $\sigma(S)$. By looking at the type in the first two bits of an encoded triple we know the length of the encoded triple. From this we can extract $\sigma(t_i)$ as well as know where $\sigma(t_{i+1})$ begins.

2. We now wish to decode each cycle to get the row and two columns of every triple, so do the following for each $\mathsf{head}$ $t_i$:

   (a) $t \leftarrow t_i$

   (b) Look at the first two bits in $t$ to determine its type, and while $t$ is not a $\mathsf{last}$, do:

      i. If $t$ is a $\mathsf{head}$, the row and two columns are stored explicitly.

      ii. If $t$ is a $\mathsf{mid}$, the two columns are stored explicitly. From the reference to $t$'s predecessor, we can calculate the diagonal shared between $t$ and its predecessor. This is can be done as we have stored which endpoint (left or right) of the predecessor lies on the diagonal, and as we decode in the same order as we encoded, we have already decoded the predecessor. We know which of $t$'s columns the shared diagonal intersects, and so we can calculate the row.

      If the predecessor is a $\mathsf{head}$, take note of which diagonal is shared with it.

      iii. $t \leftarrow$ the triple that has $t$ as its predecessor.

   (c) $t$ is now a $\mathsf{last}$, so we know the predecessor's index as well as a column. However, as the graph consists of cycles rather than just linked lists, $\mathsf{last}$ shares a diagonal with its predecessor as well as sharing a diagonal with $\mathsf{head}$. We have noted which diagonal has already been used for $\mathsf{head}$, so the other diagonal $\mathsf{head}$ touches is shared with $\mathsf{last}$.

   From these two diagonals and the column, it is possible to calculate the row by the intersection between the predecessor diagonal and the column, and the other column by the intersection of the row and the other diagonal.

3. Finally order the triples as they were, when the encoded versions were extracted in step 1.

□

# 3 Lower Bound for $N$ Vectors

In this section, we generalize the result of Section 2 to obtain a lower bound for preserving all pairwise distances amongst a set of $N$ vectors. Our proof uses Theorem 4 as a building block. Recall that Theorem 4 guarantees that there is a vector $x$ such that

$$\Pr\left[\left|\left\|\frac{1}{\sqrt{m}}TDx\right\|_2^2 - 1\right| > \varepsilon\right] \geq 2^{-\mathcal{O}(\varepsilon\sqrt{m})}.$$

Moreover, the vector $x$ has non-zeroes only in the first $O(\sqrt{m})$ coordinates. From such a vector $x$, define $x_{\to i}$ as the vector having its $j$'th coordinate equal to the $(j-i)$'th coordinate of $x$ if $j > i$ and otherwise its $j$'th coordinate is 0. In words, $x_{\to i}$ is just $x$ with all coordinates *shifted* by $i$.

For $i \leq n - O(\sqrt{m})$ ($i$ small enough that all the non-zero coordinates of $x$ stay within the $n$ coordinates), we see that $TDx$ and $TDx_{\to i}$ have the exact same distribution. Furthermore, if $i \geq m + C\sqrt{m}$ for a big enough constant $C$, $TDx$ and $TDx_{\to i}$ are independent. To see this, observe that the only random variables amongst $t_{-(m-1)}, \dots, t_{n-1}$ that are multiplied with a non-zero coordinate of $x$ in $TDx$ are $t_{-(m-1)}, \dots, t_{O(\sqrt{m})}$. Similarly, only random variables $d_1, \dots, d_{O(\sqrt{m})}$ amongst $d_1, \dots, d_n$ are multiplied by a non-zero coordinate of $x$. For $x_{\to i}$, the same is true for variables $t_{-(m-1)+i}, \dots, t_{O(\sqrt{m})+i}$ and $d_{i+1}, \dots, d_{i+O(\sqrt{m})}$. If $i \geq m + C\sqrt{m}$, these sets of variables are disjoint and hence $TDx$ and $TDx_{\to i}$ are independent.

With the observations above in mind, we now define a set of vectors $X$ as follows

$$X := \{\mathbf{0}, x, x_{\to m+C\sqrt{m}}, x_{\to 2(m+C\sqrt{m})}, \dots, x_{\to \lfloor (n-C\sqrt{m})/(m+C\sqrt{m}) \rfloor (m+C\sqrt{m})}\}.$$

The $\mathbf{0}$-vector clearly maps to the $\mathbf{0}$-vector when using $\frac{1}{\sqrt{m}}TD$ as embedding. Furthermore, by the arguments above, the embeddings of all the remaining vectors are independent and have the same distribution. It follows that

$$\begin{aligned}\Pr\left[\forall x_{\to i} \in X : \left| \left\| \frac{1}{\sqrt{m}} TDx_{\to i} \right\|_2^2 - 1 \right| \leq \varepsilon \right] &\leq \left(1 - 2^{-\mathcal{O}(\varepsilon\sqrt{m})}\right)^{|X|} \\ &\leq \exp\left(-|X| 2^{-\mathcal{O}(\varepsilon\sqrt{m})}\right).\end{aligned}$$

Now since $\mathbf{0} \in X$, it follows that to preserve all pairwise distances amongst vectors in $X$ to within $(1 \pm \varepsilon)$, we also have to preserve all norms to within $\pm\varepsilon$. This is true since for all $x$ of unit norm:

$$\left| \left\| \frac{1}{\sqrt{m}} TDx \right\|_2^2 - 1 \right| = \left| \left\| \frac{1}{\sqrt{m}} TDx - \frac{1}{\sqrt{m}} TD\mathbf{0} \right\|_2^2 - \|x - \mathbf{0}\|_2^2 \right|.$$

This proves the following:

**Theorem 10.** *Let $T$ be $m \times n$ Toeplitz and $D$ $n \times n$ diagonal. If $t_{-(m-1)}, t_{-(m-2)}, \dots, t_{n-1}$ and $d_1, \dots, d_n$ are independently distributed Rademacher random variables for $i = -(m-1), \dots, n-1$ and $j = 1, \dots, n$, then for all $0 < \varepsilon < C$, where $C$ is a universal constant, there exists a set of $N = \Omega(n/m)$ vectors $X \subset \mathbb{R}^n$ such that*

$$\Pr\left[\forall x, y \in X : \left| \left\| \frac{1}{\sqrt{m}} TDx - \frac{1}{\sqrt{m}} TDy \right\|_2^2 - \|x - y\|_2^2 \right| \leq \varepsilon \|x - y\|_2^2 \right] \leq \exp\left(-N 2^{-\mathcal{O}(\varepsilon\sqrt{m})}\right).$$

It follows from Theorem 10 that if we want to have constant probability of successfully embedding *any* set of $N$ vectors, then either it must be the case that $m = \Omega(n/N)$, or

$$-N 2^{-\mathcal{O}(\varepsilon\sqrt{m})} \geq -C_0,$$

where $C_0$ is a constant. This in turn implies that

$$\begin{aligned}\lg N - \mathcal{O}(\varepsilon\sqrt{m}) &\leq \lg C_0 &&\implies \\ \sqrt{m} &= \Omega(\varepsilon^{-1} \lg N) &&\implies \\ m &= \Omega(\varepsilon^{-2} \lg^2 N).\end{aligned}$$

This completes the proof of Theorem 3.

# References

[1] Dimitris Achlioptas. Database-friendly random projections: Johnson–Lindenstrauss with binary coins. *Journal of computer and System Sciences*, 66(4):671–687, June 2003. doi:10.1016/S0022-0000(03)00025-4.

[2] Nir Ailon and Bernard Chazelle. The fast Johnson–Lindenstrauss transform and approximate nearest neighbors. *SIAM Journal on Computing*, 39(1):302–322, May 2009. doi:10.1137/060673096.

[3] Nir Ailon and Edo Liberty. Fast dimension reduction using rademacher series on dual BCH codes. *Discrete & Computational Geometry*, 42(4):615–630, 2009. doi:10.1007/s00454-008-9110-x.

[4] Nir Ailon and Edo Liberty. An almost optimal unrestricted fast Johnson–Lindenstrauss transform. *ACM Trans. Algorithms*, 9(3):21:1–21:12, June 2013. doi:10.1145/2483699.2483701.

[5] Jeremiah Blocki, Avrim Blum, Anupam Datta, and Or Sheffet. The Johnson–Lindenstrauss transform itself preserves differential privacy. In *Proceedings of the 2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*, FOCS '12, pages 410–419, Washington, DC, USA, 2012. IEEE Computer Society. doi:10.1109/FOCS.2012.67.

[6] C. Boutsidis, A. Zouzias, M. W. Mahoney, and P. Drineas. Randomized dimensionality reduction for $k$-means clustering. *IEEE Transactions on Information Theory*, 61(2):1045–1062, February 2015. doi:10.1109/TIT.2014.2375327.

[7] E. J. Candes, J. Romberg, and T. Tao. Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52(2):489–509, February 2006. doi:10.1109/TIT.2005.862083.

[8] Michael B. Cohen, Sam Elder, Cameron Musco, Christopher Musco, and Madalina Persu. Dimensionality reduction for k-means clustering and low rank approximation. In *Proceedings of the Forty-seventh Annual ACM Symposium on Theory of Computing*, STOC '15, pages 163–172, New York, NY, USA, 2015. ACM. doi:10.1145/2746539.2746569.

[9] Anirban Dasgupta, Ravi Kumar, and Tamás Sarlos. A sparse Johnson–Lindenstrauss transform. In *Proceedings of the Forty-second ACM Symposium on Theory of Computing*, STOC '10, pages 341–350, New York, NY, USA, 2010. ACM. doi:10.1145/1806689.1806737.

[10] Sanjoy Dasgupta and Anupam Gupta. An elementary proof of a theorem of Johnson and Lindenstrauss. *Random Struct. Algorithms*, 22(1):60–65, 2003. doi:10.1002/rsa.10073.

[11] D. L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, April 2006. doi:10.1109/TIT.2006.871582.

[12] Sariel Har-Peled, Piotr Indyk, and Rajeev Motwani. Approximate nearest neighbor: Towards removing the curse of dimensionality. *Theory of Computing*, 8(14):321–350, 2012. doi:10.4086/toc.2012.v008a014.

[13] Aicke Hinrichs and Jan Vybíral. Johnson–Lindenstrauss lemma for circulant matrices. *Random Structures & Algorithms*, 39(3):391–398, 2011. doi:10.1002/rsa.20360.

[14] Piotr Indyk. Algorithmic applications of low-distortion geometric embeddings. In *Proceedings of the 42nd IEEE Symposium on Foundations of Computer Science*, FOCS '01, pages 10–33, Washington, DC, USA, 2001. IEEE Computer Society. doi:10.1109/SFCS.2001.959878.

[15] William B Johnson and Joram Lindenstrauss. Extensions of lipschitz mappings into a hilbert space. *Contemporary mathematics*, 26:189–206, 1984. doi:10.1090/conm/026/737400.

[16] Daniel M. Kane and Jelani Nelson. Sparser Johnson–Lindenstrauss transforms. *J. ACM*, 61(1):4:1–4:23, January 2014. doi:10.1145/2559902.

[17] Felix Krahmer and Rachel Ward. New and improved Johnson–Lindenstrauss embeddings via the Restricted Isometry Property. *SIAM J. Math. Anal.*, 43(3):1269–1281, 2011.

[18] Kasper Green Larsen and Jelani Nelson. Optimality of the Johnson–Lindenstrauss lemma. In *Proceedings of the 2017 IEEE 58th Annual Symposium on Foundations of Computer Science*, FOCS '17, Washington, DC, USA, October 2017. IEEE Computer Society.

[19] S. Muthukrishnan. *Data Streams: Algorithms and Applications*, volume 1(2) of *Foundations and Trends$^{TM}$ in Theoretical Computer Science.* now Publishers Inc., Hanover, MA, USA, January 2005. doi:10.1561/0400000002.

[20] Daniel A. Spielman and Nikhil Srivastava. Graph sparsification by effective resistances. *SIAM J. Comput.*, 40(6):1913–1926, December 2011. doi:10.1137/080734029.

[21] Santosh S. Vempala. *The random projection method*, volume 65 of *DIMACS - Series in Discrete Mathematics and Theoretical Computer Science.* American Mathematical Society, Providence, RI, USA, September 2004. doi:10.1007/978-1-4615-0013-1_16.

[22] Ky Vu, Pierre-Louis Poirion, and Leo Liberti. Using the Johnson–Lindenstrauss lemma in linear and integer programming. *ArXiv e-prints*, July 2015. arXiv:1507.00990.

[23] Jan Vybíral. A variant of the Johnson–Lindenstrauss lemma for circulant matrices. *Journal of Functional Analysis*, 260(4):1096–1105, 2011. doi:10.1016/j.jfa.2010.11.014.

[24] Kilian Weinberger, Anirban Dasgupta, John Langford, Alex Smola, and Josh Attenberg. Feature hashing for large scale multitask learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 1113–1120, New York, NY, USA, 2009. ACM. doi:10.1145/1553374.1553516.

[25] David P. Woodruff. *Sketching as a Tool for Numerical Linear Algebra*, volume 10(1–2) of *Foundations and Trends$^{TM}$ in Theoretical Computer Science.* now Publishers Inc., Hanover, MA, USA, 2014. doi:10.1561/0400000060.