Barriers for Faster Dimensionality Reduction

Ora Nova Fandina * Mikael Møller Høgsgaard† Kasper Green Larsen‡

Abstract

The Johnson-Lindenstrauss transform allows one to embed a dataset of n points in $\mathbb{R}^d$ into $\mathbb{R}^m$, while preserving the pairwise distance between any pair of points up to a factor $(1 \pm \varepsilon)$, provided that $m = \Omega(\varepsilon^{-2} \lg n)$. The transform has found an overwhelming number of algorithmic applications, allowing to speed up algorithms and reducing memory consumption at the price of a small loss in accuracy. A central line of research on such transforms, focus on developing fast embedding algorithms, with the classic example being the Fast JL transform by Ailon and Chazelle. All known such algorithms have an embedding time of $\Omega(d \lg d)$, but no lower bounds rule out a clean $O(d)$ embedding time. In this work, we establish the first non-trivial lower bounds (of magnitude $\Omega(m \lg m)$) for a large class of embedding algorithms, including in particular most known upper bounds.

*fandina@gmail.com. Aarhus University. Supported by Independent Research Fund Denmark (DFF) Sapere Aude Research Leader grant No 9064-00068B.

†hogsgaard@cs.au.dk. Aarhus University. Supported by Independent Research Fund Denmark (DFF) Sapere Aude Research Leader grant No 9064-00068B.

‡larsen@cs.au.dk. Aarhus University. Supported by Independent Research Fund Denmark (DFF) Sapere Aude Research Leader grant No 9064-00068B.

1 Introduction

Working with high dimensional data can be both costly in memory and computational power, motivating the study of dimensionality reduction techniques. The goal of dimensionality reduction is to take a high dimensional dataset X and embed it to a dataset Y in a lower dimensional space. If Y approximately preserves similarities between points in X, then one may use Y as input to an algorithm in place of X to save both memory and computation time at the cost of a small inaccuracy in ones output. A greatly celebrated dimensionality reduction result is the Johnson-Lindenstrauss lemma [18], which states: For any fixed $X \subset \mathbb{R}^d$, with the size of X being n, and any distortion $0 < \varepsilon < 1$, there exists a map $f : X \to \mathbb{R}^m$ such that for all $x, y \in X$

$$\|f(x) - f(y)\|_2 \in (1 \pm \varepsilon)\|x - y\|_2,$$

with m being $\Theta(\varepsilon^{-2} \lg n)$ [18, 25]. Thus the mapping is approximately preserving the Euclidean distances between the points in X in the lower dimensional space $\mathbb{R}^m$. The property of preserving pairwise distances via the Johnson-Lindenstrauss lemma have found great use in many applications, for instance as a preprocessing step to speed up machine learning algorithms.

A standard approach for obtaining an embedding f satisfying the above, is to pick a random $m \times d$ matrix A with each entry being i.i.d. $N(0, 1)$ distributed [15] (or uniform $-1/1$ [5]) and embedding any input $x \in X$ to $f(x) = m^{-1/2}Ax$. Computing such an embedding thus takes $O(md)$ time. In some applications of dimensionality reduction, this becomes the bottleneck in the running time, thus motivating faster embedding algorithms. The work on faster dimensionality reduction in Euclidian space can be divided roughly into two categories: 1) using sparse embedding matrices A, or 2), using matrices A with special structure that allows fast matrix-vector multiplication. In both cases, the fastest embedding algorithms use super-linear time in the input dimensionality in the worst case. For sparse matrices, there is near-tight lower bound by Nelson and Nguyen [28] showing that the embedding time cannot be reduced below roughly $\Omega(d\varepsilon^{-1} \lg n)$. For structured matrices, the fastest embeddings use at least $\Omega(d \lg m)$ time, however in this case there are no lower bounds ruling out faster embeddings that could conceivably embed a vector in $O(d)$ time see e.g. [10, 17]. Working towards such lower bounds is the focus of this work.

Our Contributions. In this work, we establish the first non-trivial lower bounds on the time required for dimensionality reduction in Euclidian space when not restricted to using sparse matrices to perform the embedding. Focusing on the case of $d = cm$, for a constant $c > 1$ and optimal $m = O(\varepsilon^{-2} \lg n)$, we prove that a *large class* of embedding algorithms, including most known upper bounds, must use time $\Omega(m \lg m)$. This coincides with known upper bounds for several tradeoffs between ε and n. In addition to establishing a first lower bound, we believe our careful definition of the class of algorithms that the lower bound applies to, shines light on the barriers faced when developing fast embedding algorithms.

In the following section, we survey previous work and formally present our results.

1.1 Fast Dimensionality Reduction

As mentioned above, the previous work on fast dimensionality reduction can be divided into two categories, either based on sparse matrices or on structured matrices. We elaborate on these approaches in the following.

Sparse JL. The basic idea in sparse JL embeddings, is to use an embedding matrix A with only $s < m$ non-zeros per column. With such a matrix A, the product Ax can be computed trivially in $O(sd)$ time rather than $O(md)$, thus speeding up the embedding. Moreover, if x itself has few non-zeros, then the product may even be computed in $O(s\|x\|_0)$ time, where $\|x\|_0$ is the number of non-zeros in x. Using sparse embedding matrices was initiated by [1] and culminated with the current state-of-the-art embedding by Kane and Nelson [21] who showed that it suffices to pick a matrix A having $s = O(\varepsilon^{-1} \lg n)$ random entries (without replacement) in each column set uniformly and independently to $-1/1$ and embedding a vector x to $s^{-1/2}Ax$. Moreover, this nearly matches a sparsity lower bound by Nelson and Nguyen [28] who showed that any sparse embedding matrix must have $s = \Omega(\varepsilon^{-1} \lg n / \lg(1/\varepsilon))$ non-zeros per column. Another line

of research in this direction, studies sparsities s below the lower bound by Nelson and Nguyen. For instance, Feature Hashing [33] considers the extreme case of $s = 1$. Of course, such embeddings cannot work for all data sets X. However, as shown by Weinberger et al. [33] and later refined by Kamma et al. [11] and generalized to $s > 1$ by Jagadeesan [16], one can use extremely sparse embedding matrices, provided that for all pairwise difference vectors $z = x - y$ for $x, y \in X$, the ratio $\|z\|_\infty / \|z\|_2$ is small. That is, there are no single large coordinates in z.

Fast JL. The second line of research on fast embeddings exploits structured matrices A with fast matrix-vector multiplication algorithms. Ailon and Chazelle [2] initiated this direction by introducing the FastJL transform. FastJL embeds a vector by computing a product $m^{-1/2}PHDx$, where P is a sparse matrix, H is the normalized $d \times d$ Hadamard matrix and D is a diagonal matrix with random signs on the diagonal. The trick is that computing Dx can be done in $O(d)$ time and computing $H(Dx)$ takes only $O(d \lg d)$ time by exploiting the structure of the Hadamard matrix. Finally, the transformation HDx has the effect of "smoothening" out the coordinates of the input vector, making the ratio $\|HDx\|_\infty / \|HDx\|_2$ small. This is precisely the setup allowing very sparse embedding matrices. Concretely, Ailon and Chazelle [2] showed that it suffices to let each entry in P be non-zero with probability $q = O((\lg^2 n)/d)$, resulting in a total embedding time of $O(d \lg d + m \lg^2 n)$. Their analysis was recently refined by Fandina et al. [10], showing that the sparsity parameter q can be reduced further. Numerous other embeddings exploiting structured matrices has since then been introduced [22, 8, 3, 6], including for instance embeddings based on Toeplitz matrices [14, 31, 12] and the Kac random walk [19, 17]. If one insists on optimal $m = O(\varepsilon^{-2} \lg n)$ dimensions in the embedding, then the current state-of-the-art is either the FastJL transform or the Kac random walk depending on the relationship between n and ε. However none of these are faster than $O(d \lg m)$ for any tradeoff between ε and n.

Unlike the sparse matrix case, there are no known lower bounds ruling out e.g. $O(d)$ time embeddings via structured matrices. Naturally, the reason for this, is that it is much harder to prove lower bounds for general embedding algorithms that exploit structured matrices than merely bounding the sparsity of the embedding matrix. In fact, proving super-linear lower bounds for general linear circuits (which capture current embedding algorithms) is a major open question in complexity theory. In light of this obstacle, which we will elaborate on in Section 1.3, we identify common traits in most known upper bounds that we exploit to prove lower bounds for dimensionality reduction. In the following, we formally define the model under which we prove our lower bound.

1.2 Formal Lower Bound

As mentioned earlier, our lower bound holds for a large class of dimensionality reducing maps. This class is captured by a certain scaling parameter. Concretely, we define a ScaledJL-matrix as follows:

Definition 1. *Let $0 < \varepsilon, \delta < 1$ and $s \in \mathbb{N}$. A stochastic matrix $A \in \mathbb{R}^{m \times d}$ is said to be a ScaledJL(ε, δ, s)-matrix, if for any $x \in \mathbb{R}^d$ we have that*

$$\mathbb{P}_A \left[\left\| s^{-1/2} Ax \right\|_2^2 \notin (1 \pm \varepsilon) \|x\|_2^2 \right] < \delta.$$

Let us remark a few things about Definition 1. First, we assume that a ScaledJL(ε, δ, s)-matrix $s^{-1/2}A$ preserves the (squared) norm of any single vector x up to $(1 \pm \varepsilon)$ except with probability δ. This is the standard definition of a distributional Johnson-Lindenstrauss transform and all known upper bounds give such a guarantee. In greater detail, known upper bounds prove the distributional guarantee and then sets $\delta < 1/n^2$ and applies a union bound over all $z = x - y$ for $x, y \in X$ to conclude that the embedding preserves all pairwise (squared) distances among vectors in X. In this work, we focus on the squared distance as it simplifies calculations and anyways only changes ε by a constant factor. The non-standard thing in Definition 1 is the scaling parameter s. Of course, such a scaling parameter can also be implicitly hidden in A by scaling all entries of A by $s^{-1/2}$. To explain the role of s in our model, we need to first introduce a linear circuit/algorithm as defined e.g. by Morgenstern:

Definition 2. *[26] A linear algorithm takes as an input $1, x_1, \ldots, x_d \in \mathbb{R}$ and proceeds in $t > 0$ steps. In the l'th step the algorithm computes x_{d+l} by $x_{d+l} = \lambda_{d+l} x_j + \mu_{d+l} x_i$ for some pair of indices $i, j < d + l$, where $\lambda_{d+l}, \mu_{d+l} \in \mathbb{R}$.*

We say that a linear algorithm computes a linear transformation $B \in \mathbb{R}^{m \times d}$ if there exist indices $1 \leq k_1, \ldots, k_m \leq d + t$ such that: $(Bx)_1 = x_{k_1}, \ldots, (Bx)_m = x_{k_m}$ for every possible input $x = (x_1, \ldots, x_d) \in \mathbb{R}^d$.

Note that the number of steps t determines the number of operations performed by the algorithm (up to a factor 3). Proving super-linear lower bounds for linear algorithms in the sense of Definition 2, is a major open problem [30]. Thus several previous works [27, 7] have considered restrictions where the coefficients λ and μ are bounded in absolute value by a constant r independent of m and d. This is crucially necessary if one wants to avoid the long-standing complexity theoretic barriers further elaborated on in Section 1.3.

With this in mind, the role of s in our definition of ScaledJL(ε, δ, s)-matrix becomes clearer. Concretely, if we consider an embedding $s^{-1/2}Ax$, then we think of A as being computable by a linear algorithm/circuit where all coefficients λ_i and μ_i are bounded by a constant. This naturally leads to a scaling factor $s^{-1/2}$ for some s. Such a scaling also occurs in most known upper bounds. Let us first state our main lower bound result and then discuss how it relates to known constructions:

Theorem 3. *Let $A \in \mathbb{R}^{m \times d}$ be a ScaledJL(ε, δ, s)-matrix for $\varepsilon \leq 1/4$, $\delta \leq C$ (C being some universal constant), $s \in \mathbb{N}$, $m = \Theta(\varepsilon^{-2} \lg(1/\delta))$ and $d \geq m$, then the expected (over the random choice of A) minimum number of operations needed for any linear algorithm computing A with $|\lambda_i|, |\mu_i| \leq 1$ for all i is $\Omega(m \lg s)$.*

Let us briefly argue that most known constructions are of the form captured by the lower bound and the definition of a ScaledJL(ε, δ, s)-matrix. Concretely, these upper bounds have $\lg s = \Omega(\lg m)$ and thus our lower bound shows that it must take $\Omega(m \lg m)$ operations to compute these embeddings, even if more clever linear algorithms could be devised. As an example of an upper bound, consider first the classic JL construction using a matrix A with i.i.d. random $-1/1$ entries and a scaling of $s^{-1/2} = m^{-1/2}$. In this case, the matrix A can clearly be computed by a linear algorithm using coefficients bounded by 1 in absolute value (just carry out the trivial algorithm). So it falls under the definition of a ScaledJL(ε, δ, s)-matrix with $s = m$. Next consider embeddings based on Toeplitz matrices [14, 31, 12]. Here we embed as $m^{-1/2}TDx$, where D is a diagonal with random signs and T is a Toeplitz matrix with random signs on its diagonals. The matrix T can be computed via a fast Fourier transform using coefficients bounded by a constant. Hence the construction also falls under the definition of ScaledJL(ε, δ, s)-matrix with $s = m$. We could also consider the sparse JL transform by Kane and Nelson [21]. Their construction uses an embedding matrix where each column has $t = \Theta(\varepsilon^{-1} \lg n)$ non-zero entries, each of magnitude $t^{-1/2}$. Such a sparse embedding is typically computed by moving the scaling $t^{-1/2}$ outside and then doing the straight-forward sparse matrix-vector multiplication using constant magnitude coefficients. It thus falls under the definition of a ScaledJL(ε, δ, s)-matrix with $s = t = \Theta(\varepsilon m)$. This has $\lg s = \Omega(\lg m)$ when m is optimal $O(\varepsilon^{-2} \lg n)$. Finally, consider for instance the $m^{-1/2}PHD$ construction by Ailon and Chazelle [2]. They use the *normalized* Hadamard matrix, i.e. all entries in H are scaled down by $d^{-1/2}$. If we move that scaling factor outside, as $(md)^{-1/2}P\bar{H}D$, then $\bar{H}$ is computed recursively using coefficients of 1 and -1. The entries of P are $b \cdot N(0, q^{-1})$ distributed, where b is a Bernoulli random variable with success probability q for a $q > \lg(1/\delta)/d$. With high probability, no entry of P is thus larger than about $O(\sqrt{d})$. Moving this scaling factor outside, it cancels out with the $d^{-1/2}$ from the Hadamard matrix and then P can also be computed using coefficients bounded by a constant and the final algorithm is a ScaledJL(ε, δ, s)-matrix with $s = \Theta(m)$. Common to all these approaches, is that they project onto something that resembles a random m-dimensional subspace. Intuitively, such a matrix should have m rows all of norm about $\sqrt{d/m}$. With d columns, this would imply that each entry should be about $m^{-1/2}$ in magnitude. Moving the scaling factor outside to have constant magnitude entries, results in the $m^{-1/2}$ scaling factor observed in all these upper bounds.

Thus many known upper bounds fall under the definition of a ScaledJL(ε, δ, s)-matrix with a scaling s satisfying $\lg s = \Omega(\lg m)$. Theorem 3 therefore sheds light on why they all require $\Omega(m \lg m)$ time (which is $\omega(d)$ when $d = O(m)$). Let us also mention the only upper bound we are aware of, that does not seem to suffer from the lower bound. In the Kac JL transform [19, 17], one embeds a vector by repeatedly picking two random coordinates, among the d input coordinates, and performing a random rotation on the two.

After sufficiently many steps ($\Omega(d \lg d + m \lg n)$ in the current analysis), all but the first m coordinates are discarded and those m coordinates are scaled by $\sqrt{d/m}$. While seemingly not being captured by the lower bound, we remark that the analysis of Kac JL cannot be sharpened to $o(d \lg d)$ steps as otherwise, by a coupon collector argument, there is a vector e_i among $e_{m+1}, \ldots, e_d$ whose coordinate i is never involved in a rotation and hence e_i is embedded to 0.

Of course, it would have been more natural, if our lower bound in Theorem 3 only required bounded coefficients in the linear algorithm, not that there is also a scaling parameter $s^{-1/2}$. Unfortunately, as we argue in Section 1.3, it seems unlikely that we can establish such a lower bound using current techniques. We thus believe our results can be seen in two ways: 1), as providing strong evidence that FastJL constructions cannot be made much faster, or 2), as pointing towards a direction for further improvements, by trying to design embeddings where a constant scaling parameter s suffices, or super-constant coefficients are used when computing the embedding, or perhaps using non-linearity.

1.3 Barriers for Linear Algorithm Lower Bounds

Proving super-linear unconditional lower bounds is one of the biggest barriers in many areas of complexity theory, including in particular for linear operators. A natural computational model for computing linear operators is a linear algorithm, a.k.a. linear circuit, as in Definition 2. While being a very natural model of computation for linear operators, capturing in particular all known JL constructions, it suffers from a lack of tools for proving lower bounds (without any assumptions on coefficients). Concretely, there are still no super-linear size lower bounds, even for circuits restricted to logarithmic depth. Moreover, this road block is not for lack of trying. For instance, already in 1977, Valiant [30] introduced the notion of *matrix rigidity*. Loosely stated, the rigidity of a square matrix (corresponding to a linear operator) $A \in \mathbb{R}^{n \times n}$, is the minimum number of entries in A that needs to be changed to reduce its rank below $n/2$. Valiant showed that any explicit matrix A with rigidity $\Omega(n^2/\lg \lg n)$ cannot have a linear-sized and log-depth linear circuit for computing the corresponding linear operator. Matrix rigidity has since then been the topic of much research, see e.g. [13, 4, 29, 9], however none of these works lead to super-linear lower bounds (also when considering rectangular matrices) for explicit matrices, despite the fact that a random matrix has high rigidity with high probability.

Bounded Coefficients. In light of the above strong barriers for proving lower bounds for linear circuits, a natural restriction to the computational model, is to assume that all coefficients λ_i and μ_i used by the gates are bounded in absolute value by a constant r. Indeed, if we enforce such a restriction, then Morgenstern [27] for instance proved an $\Omega(n \lg n)$ lower bound on the size of any linear circuit computing the $n \times n$ *unnormalized* fast Fourier transform. Similarly, Chazelle [7] proved $\Omega(n \lg n)$ lower bounds for linear circuits, with bounded integer coefficients, for computing linear transformation corresponding to incidence matrices for various geometric range searching problems. Common to these techniques, is that they relate the circuit complexity to the eigenvalues of the corresponding matrix A. In particular, the lower bounds one obtains peak at $\Omega(\ell \lg \gamma_\ell)$, where γ_ℓ denotes the ℓ'th largest eigenvalue of $A^T A$.

Now in the context of dimensionality reduction, an embedding matrix $A \in \mathbb{R}^{m \times d}$ can have at most m non-zero eigenvalues. This means that lower bounds obtained via these techniques will be proportional to only $\Omega(m \lg \gamma_\ell)$ for an $\ell \in \Theta(m)$. Since the size of the circuit is already at least d, it makes most sense from a lower bound point of view to consider setups where m and d are within constant factors. However, since embedding matrices A must preserve the norm of standard unit vectors e_i, their columns will have norms of magnitude $(1 \pm \varepsilon)$. This implies that the trace of $A^T A$ is $d(1 \pm \varepsilon) = \Theta(m)$. Since the trace of $A^T A$ equals the sum of its eigenvalues, we get for $\ell \in \Theta(m)$ that γ_ℓ is at best a constant. Thus the lower bounds we may hope to obtain are only $\Omega(m)$, i.e. trivial. Thus considering only the restriction to have coefficients bounded by a constant is insufficient for proving non-trivial lower bounds using known techniques.

Output Scaling. Having observed the above, we examined existing FastJL constructions and found a common trait in most of them: they embed a vector x by computing $s^{-1/2} Ax$ for some scaling factor s and

matrix A, where A can be computed efficiently by a linear circuit using coefficients of constant magnitude. Given the obstacles mentioned above, we thus settled on proving lower bounds for embeddings that follow this template, resulting in Theorem 3 above.

2 Lower Bound for Linear Algorithms

The goal of this section is to prove our lower bound from Theorem 3 on the operations needed for any linear algorithm computing a ScaledJL(ε, δ, s)-matrix. We state a stronger version of the theorem here:

Theorem 4. *Let $A \in \mathbb{R}^{m \times d}$ be a ScaledJL(ε, δ, s)-matrix for $\varepsilon \leq 1/4$, $\delta \leq C$(C being some universal constant), $s \in \mathbb{N}$ and $t\varepsilon^{-2}\lg(1/\delta) = m$, $t \geq 1$ and $d \geq m$, then the expected (over the random choice of A) minimum number of operations needed for any linear algorithm computing Ax for any $x \in \mathbb{R}^d$ with $|\lambda_i|, |\mu_i| < r$ for all i and $r > 1/2$, is $\Omega(m\lg(s/t^2)/(t\lg(2r))$.*

We notice that Theorem 3 is a special case of Theorem 4 where r is set equal to 1 and $t = \Theta(1)$.

The main tool for proving Theorem 4 is a lemma by Morgenstern relating the operations needed by a linear algorithm computing a linear transformation B, to the determinants of square submatrices of B:

Lemma 5. *[27] Let B be a real matrix and let $\Delta(B)$ denote the maximum over the absolute value of the determinant of any square submatrix of B. A linear algorithm computing the linear transformation B, with $|\lambda_i|, |\mu_i| < r$ for all i and $r > 1/2$, must use at least $\lg(\Delta(B))/\lg(2r)$ operations.*

Using Lemma 5 as our offset, our goal is thus to show that any ScaledJL(ε, δ, s)-matrix A must have a submatrix whose determinant is in the order of $s^{\Omega(m)}$. Since A is allowed to be stochastic and fail to preserve the norm of a vector x with probability δ, we only prove that this holds with constant probability over A:

Lemma 6. *Let $A \in \mathbb{R}^{m \times d}$ be a ScaledJL(ε, δ, s)-matrix for $\varepsilon \leq 1/4$, $\delta \leq C$ (C being some universal constant), $s \in \mathbb{N}$ and $t\varepsilon^{-2}\lg(1/\delta) = m$, $t \geq 1$ and $d \geq m$, then there exist a set $S \subseteq supp(A)$ such that $\mathbb{P}_A[S] \geq 1/2$ and for $B \in S$ it holds that there exists a square submatrix F of B such that*

$$|\det(F)| \geq \left(c^2 s/(3(et)^2)\right)^{\lceil cm/t \rceil/2}$$

where c is some universal constant less than 1.

The proof of Theorem 4 follows immediately from the above two lemmas:

Proof of Theorem 4. Let A be a ScaledJL(ε, δ, s)-matrix. Lemma 6 gives the existence of a set $S \subseteq \text{supp}(A)$ with $\mathbb{P}_A[S] \geq 1/2$ and for $B \in S$, B has a square submatrix F such that $|\det(F)| \geq \left(c^2 s/(3(et)^2)\right)^{\lceil cm/t \rceil/2}$ implying that $\Delta(B) \geq \left(c^2 s/(3(et)^2)\right)^{\lceil cm/t \rceil/2}$. It now follows by Lemma 5 that a linear algorithm calculating Bx for all $x \in \mathbb{R}^d$ must use $\lg(\Delta(B))/\lg(2r)$ operations. Since $\lg(\Delta(B)) \geq (\lceil cm/t \rceil)\lg(c^2 s/(3(et)^2))/2 = \Omega(m\lg(s/t^2)/t)$ we get that $\lg(\Delta(B))/\lg(2r) = \Omega(m\lg(s/t^2)/(t\lg(2r))$. Thus we conclude, since $\mathbb{P}_A[S] \geq 1/2$, that the expected number of operations needed by any linear algorithm computing the transformation A is $\Omega(m\lg(s/t^2)/(t\lg(2r))$, which concludes the proof of Theorem 4. □

The main challenge we face is thus establishing Lemma 6, i.e. proving that for any ScaledJL(ε, δ, s)-matrix A, it is often the case that A has a square submatrix of large determinant. This is the focus of the next section.

2.1 Submatrix with Large Determinant (Proof Lemma 6)

To prove Lemma 6, we have to show that with probability at least 1/2, a ScaledJL(ε, δ, s)-matrix has a square submatrix with an $\left(c^2 s/(3(et)^2)\right)^{\lceil cm/t \rceil/2}$ large determinant. For this, we will use a technical lemma from [23] which relates the eigenvalues of $B^T B$ to the determinants of square submatrices of B:

Lemma 7. *([23] proof of Theorem 10) For $B \in \mathbb{R}^{m\times d}$, with $m \leq d$, let $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m \geq 0$ denote the eigenvalues of $B^T B$. For all positive integers $l \leq m$, there exists a square submatrix $F \in \mathbb{R}^{l\times l}$ of B such that*

$$|\det(F)| \geq \sqrt{\frac{\prod_{i=1}^{l} \lambda_i}{\binom{d}{l}\binom{m}{l}}}.$$

By the above lemma, we can reduce the problem of finding a square submatrix of a ScaledJL(ε, δ, s)-matrix A with large determinant, to lower bounding the eigenvalues of a $A^T A$. Using $\lambda_i(B^T B)$ to denote the i'th largest eigenvalue of $B^T B$, this is precisely the contents of the following lemma:

Lemma 8. *Let $A \in \mathbb{R}^{m\times d}$ be a ScaledJL(ε, δ, s)-matrix for $\varepsilon \leq 1/4$, $\delta \leq C$ (C being some universal constant) and $s \in \mathbb{N}$, $t\varepsilon^{-2}\lg(1/\delta) = m$, $t \geq 1$ and $d \geq m$, then there exist a set $S \subseteq supp(A)$ such that $\mathbb{P}_A[S] \geq 1/2$ and for $B \in S$ it holds that*

$$\lambda_{\lceil cm/t \rceil}(B^T B) \geq ds/(3m)$$

where c is some universal constant less than 1.

Before we give the proof of Lemma 8, let us see that it suffices to finish the proof of Lemma 6:

Proof of Lemma 6. Let A be ScaledJL(ε, δ, s)-matrix such that the conditions of Lemma 8 are met. We then have for B in the set S described in Lemma 8 that the $l = \lceil cm/t \rceil$'th largest eigenvalue of $B^T B$ is at least $ds/(3m)$. Now by Lemma 7. we have that there exist a square submatrix $F \in \mathbb{R}^{l\times l}$ of B such that $|\det(F)| \geq (\prod_{i=1}^{l} \lambda_i / \binom{d}{l}\binom{m}{l})^{1/2}$. Now using these two properties combined with $\binom{n}{k} \leq (en/k)^k$ and $l \geq cm/t$ we get that

$$|\det(F)| \geq \left(\prod_{i=1}^{l} \lambda_i / \left(\binom{d}{l}\binom{m}{l}\right)\right)^{1/2} \geq \left(dsl^2/(3e^2 dm^2)\right)^{l/2} \geq \left(c^2 s/(3(et)^2)\right)^{\lceil cm/t \rceil/2}.$$

Thus Lemma 6 follows by the conditions in Lemma 6 and Lemma 8 on the ScaledJL(ε, δ, s)-matrix being the same. □

After having established the above connection between eigenvalues and linear algorithms, we are left with proving Lemma 8, i.e. to show that for a ScaledJL(ε, δ, s)-matrix A, it is often the case that $A^T A$ has many large eigenvalues. We first give an overview of the main ideas in the proof, before proceeding to give the formal details.

Proof Overview. The proof of Lemma 8 is at a high level inspired by methods used in [24]. The main result of [24] was a lower bound of $m = \Omega(\varepsilon^{-2}\lg n)$ on the embedding dimension of any linear dimensionality reducing map. Their lower bound was proved for a "hard" set of vectors consisting of the standard basis vectors and several independent Gaussian vectors. The standard basis vectors were used to lower bound the trace $Tr(A^T A)$ where A is the full embedding matrix (including any scaling factors), whereas the Gaussian vectors were used to upper bound the squared Frobenius norm $\|A^T A\|_F^2$. Since $Tr(A^T A)$ is the sum of the eigenvalues of $A^T A$ and $\|A^T A\|_F^2$ is the sum of squared eigenvalues, one cannot have a large $Tr(A^T A)$ and a small $\|A^T A\|_F^2$ without having many non-zero eigenvalues. Their lower bound on m follows by observing that the number of non-zero eigenvalues equals the rank of A, and the rank cannot exceed m. We remark that the idea of using Gaussian vectors as a hard instance was also seen in [20].

Compared to the proof above, we need to show something stronger. More precisely, the previous work merely showed that there are $\Omega(\varepsilon^{-2}\lg n)$ non-zero eigenvalues. We need to show that there are $\Omega(\varepsilon^{-2}\lg n)$ eigenvalues that are all at least $ds/(3m)$ large. This requires a more refined analysis and the introduction of the scaling parameter $s^{-1/2}$ in the embedding $s^{-1/2}Ax$ as in the definition of a ScaledJL(ε, δ, s)-matrix.

The hard instance in our lower bound is also the standard basis vectors $e_1, \ldots, e_d$ in $\mathbb{R}^d$ together with a Gaussian distributed vector $g \in \mathbb{R}^d$. By Markov's inequality, we get that the following two events hold

simultaneous with constant probability over the random choice of A: The number of basis vectors whose norm is preserved, i.e. $|\{i : \|Ae_i/\sqrt{s}\|^2 \in (1 \pm \varepsilon)\}|$, is $\Omega(d)$, and secondly, the probability that the random Gaussian vector has its norm preserved satisfies $\mathbb{P}_g[\|Ag/\sqrt{s}\|^2 / \in (1 \pm \varepsilon) \|g\|^2] \geq 1 - \Theta(\delta)$. Thus if we now consider an outcome B of A which satisfies these two relations, we get by $|\{i : \|Be_i/\sqrt{s}\|^2 \in (1 \pm \varepsilon)\}| = \Omega(d)$ that the trace of $B^T B$, which is equal to the sum of the eigenvalues $B^T B$, is $\Omega(ds)$. Now by $\|Bg/\sqrt{s}\|^2$ being in $(1 \pm \varepsilon) \|g\|^2$ and $\|g\|^2$ being in $(1 \pm \varepsilon)d$, both with probability least $1 - \delta^{\Theta(1)}$ over g, we also get with probability at least $1 - \delta^{\Theta(1)}$ over g that $\|Bg\|^2 \in (1 \pm \Theta(\varepsilon))ds$.

Now using the lower bound $\sum \lambda(B^T B)_i = \Omega(ds)$ and the fact that $B^T B$ has at most m non-zero eigenvalues, we get that the sum of the eigenvalues larger than $ds/(3m)$ is at least $\Omega(ds) - m(ds/(3m)) = \Omega(ds)$ (provided that we can prove a large enough constant in the $\Omega(ds)$ notation). However, we also need to prove that there are not just a few such eigenvalues that are huge and account for most of the sum. For this, let l denote the number of eigenvalues that are greater than or equal to $ds/(3m)$.

To prove a lower bound on l, we first use anti-concentration inequalities to relate the distribution of $\|Bg\|^2$ to $Tr(B^T B)$, obtaining an upper bound on $\|B^T B\|_F^2 = \sum \lambda(B^T B)_i^2 \leq O((ds)^2/m)$ (like in previous work). Using the upper bound on $\sum \lambda(B^T B)_i^2$ and Cauchy-Schwartz, we then conclude that the sum of the eigenvalues larger than $ds/(3m)$ is at most $\Theta(ds\sqrt{l/m})$ - hence combining the lower and upper bound on the sum of the eigenvalues larger than $ds/(3m)$, we get that $\Theta(ds\sqrt{l/m}) = \Omega(ds)$, so we conclude that $l = \Omega(m)$ as wanted. We remark that while this last part of our proof carries some resemblance to that in [24], we believe that the whole reduction above, reducing the problem to arguing that the embedding matrix must have many large eigenvalues, is highly novel in its own right.

Preliminaries. To prove Lemma 8, we need the following two concentration bounds for normal distributed random variables.

Lemma 9. *[34] Let $g_1, \ldots, g_d$ be independent $N(0,1)$ random variables and $u_1, \ldots, u_d$ be non-negative numbers, then for constants $c_1 \leq 1$ and $C_1 \geq 1$ we have that*

$$c_1 \exp\left(-C_1 x^2/\|u\|_2^2\right) \leq \mathbb{P}\left[\sum_{i=1}^{d} u_i(g_i^2 - 1) \geq x\right], \quad \forall 0 \leq x$$

$$c_1 \exp\left(-C_1 x^2/\|u\|_2^2\right) \leq \mathbb{P}\left[\sum_{i=1}^{d} u_i(g_i^2 - 1) \leq -x\right], \quad \forall 0 \leq x \leq c_1\|u\|_2^2/\|u\|_\infty.$$

Lemma 10. *(Example 2.11 [32]) Let $g_1, \ldots, g_d$ be independent $N(0,1)$ random variables then*

$$\mathbb{P}\left[\left|\sum_{k=1}^{d} g_i^2 - d\right| \geq \alpha d\right] \leq 2e^{-d\alpha^2/8}, \quad \text{for all } \alpha \in (0,1).$$

Proof of Lemma 8. We are now ready to give the proof of Lemma 8.

Proof. Let $A \in \mathbb{R}^{m \times d}$ be a ScaledJL(ε, δ, s)-matrix for $\varepsilon \leq 1/4$ and $\delta \leq C$ (where C is a constant to be fixed later), $t\varepsilon^{-2}\lg(1/\delta) = m$ and $d \geq m$.

Let $e_1, \ldots, e_d$ be the standard basis vectors in $\mathbb{R}^d$. Let further $\mathbb{P}_g$ denote the measure of a standard Gaussian random vector $g \in \mathbb{R}^d$ independent of A. We now claim the existence of a set of matrices S such that $A \in S$ holds with probability at least $1/2$ and for $B \in S$, we have that

$$|\{i : |\left\|Be_i/\sqrt{s}\right\|^2 - \|e_i\|^2| > \varepsilon \|e_i\|^2\}| < 4\delta d \tag{1}$$

and

$$\mathbb{P}_g\left[|\left\|Bg/\sqrt{s}\right\|^2 - \|g\|^2| > \varepsilon \|g\|^2\right] < 4\delta. \tag{2}$$

To show this, define for each $i \in [d]$ the event $E_i = \{|\, \|Ae_i/\sqrt{s}\|^2 - \|e_i\|^2 \,| > \varepsilon \|e_i\|^2\}$ and set X_i equal to $\mathbf{1}_{E_i}$, such that $\sum_{i=1}^d X_i = |\{i : |\, \|Be_i/\sqrt{s}\|^2 - \|e_i\|^2 \,| > \varepsilon \|e_i\|^2\}|$. By the ScaledJL$(\varepsilon, \delta, s)$-matrix assumption of A, we have that

$$\mathbb{E}_A \left[\sum_{i=1}^d X_i \right] \leq \delta d$$

so by Markov's inequality we get that

$$\mathbb{P}_A \left[\sum_{i=1}^d X_i \geq 4\delta d \right] \leq 1/4$$

similarly by the ScaledJL(ε, δ, s)-matrix assumption we have that

$$\mathbb{E}_A \left[\mathbb{P}_g \left[|\, \left\|Ag/\sqrt{s}\right\|^2 - \|g\|^2 \,| > \varepsilon \|g\|^2 \right] \right] < \delta$$

so by applying Markov's inequality again, we get that

$$\mathbb{P}_A \left[\mathbb{P}_g \left[|\, \left\|Ag/\sqrt{s}\right\|^2 - \|g\|^2 \,| > \varepsilon \|g\|^2 \right] \geq 4\delta \right] \leq 1/4.$$

Now using a union bound gives that eq. (1) and eq. (2) hold simultaenously with probability at least $1/2$ as claimed.

If we can show that for $B \in S$, it holds that $\lambda(B^T B)_{\lceil cm/t \rceil} > ds/(3m)$, then we are done since the probability of A being in S is at least $1/2$. So let $B \in S$. We now notice that by eq. (1) there exist $(1-4\delta)d$ indices in $i \in [d]$ such that $(B^T B)_{i,i} \in (1 \pm \varepsilon)s$. If we now let $\lambda_i(B^T B)$ denote the i'th largest eigenvalue of $B^T B$, we get the following lower bound on the sum of eigenvalues of $B^T B$ (assuming $\varepsilon \leq 1/4$ and $\delta \leq C \leq 1/36$):

$$\sum_{i=1}^m \lambda_i(B^T B) = Tr(B^T B) \geq (1-\varepsilon)(1-4\delta)ds \geq 2ds/3. \tag{3}$$

Now by Cauchy-Schwartz, we also have that

$$\sum_{i=1}^m \lambda_i(B^T B) \leq \sqrt{m \sum_{i=1}^m \lambda_i(B^T B)^2} \leq \sqrt{m \sum_{i=1}^m \lambda_i(B^T B)^2} \sqrt{\sum_{i=1}^m \lambda_i(B^T B)^2/\lambda_1} = \sqrt{m} \sum_{i=1}^m \lambda_i(B^T B)^2/\lambda_1 \tag{4}$$

Combining eq. (3) and eq. (4), we get that

$$(\sum_{i=1}^m \lambda_i(B^T B)^2)/\lambda_1(B^T B) \geq 2ds/3\sqrt{m} \geq ds/4\sqrt{m}. \tag{5}$$

Now since B was in S, we have by eq. (2) that $\|Bg\|^2 \in (1 \pm \varepsilon)\|g\|^2$ with probability at least $1 - 4\delta$ over g. At the same time, we have by Lemma 10 that for $0 < \alpha < 1$, it holds that $\|g\|^2 \in (1 \pm \alpha)d$ with probability at least $1 - 2\exp(-d\alpha^2/8)$. Now choosing $\alpha = \varepsilon$, we get that $2\exp(-d\varepsilon^2/8) \leq 2\delta^{1/8}$. By the assumption that $d \geq m \geq \varepsilon^{-2} \lg(1/\delta)$, we get that $\|g\|^2 \in (1 \pm \varepsilon)d$ with probability at least $1 - 2\delta^{1/8}$ over g. Now combining this with $\|Bg\|^2 \in (1 \pm \varepsilon)\|g\|^2$ with probability at least $1 - 4\delta$ over g, we get by a union bound that

$$\|Bg\|^2 \in (1 \pm \varepsilon)(1 \pm \varepsilon)ds = (1 - 2\varepsilon + \varepsilon^2, 1 + 2\varepsilon + \varepsilon^2)ds \tag{6}$$

with probability at least $1 - 6\delta^{1/8}$ over g.

Now using the eigenvalue decomposition of B^TB into U^TDU, where U is an orthogonal matrix and D an diagonal matrix with the eigenvalues of B^TB on its diagonal in decreasing order, and that a standard normal Gaussian vector is invariant in distribution under rotations, we obtain the following relation

$$\begin{aligned} \|Bg\|^2 - Tr(B^TB) = \\ g^TB^TBg - Tr(B^TB) = \\ g^TU^TDUg - Tr(B^TB) \stackrel{d}{=} \\ \tilde{g}^TD\tilde{g} - \sum_{i=1}^{d}\lambda_i(B^TB) = \\ \sum_{i=1}^{d}\lambda_i(B^TB)(\tilde{g}_i^2 - 1). \end{aligned} \tag{7}$$

Our next step is to relate $\sum_i \lambda_i^2(B^TB)$ to δ. Here we take two different approaches depending on $Tr(B^TB)$. c_1 and C_1 in the following are the constants of Lemma 10.

Case 1: If $Tr(B^TB) \leq (1 - 2\varepsilon + c_1/(4\sqrt{m}))ds$ then by eq. (6) (and the comment above the equation) we have with probability at least $1 - 6\delta^{1/8}$ over g that

$$\|Bg\|^2 - Tr(B^TB) \geq ((1 - 2\varepsilon + \varepsilon^2) - (1 - 2\varepsilon + c_1/(4\sqrt{m})))ds > -c_1ds/4\sqrt{m}.$$

implying that $6\delta^{1/8} \geq \mathbb{P}_g\left[\|Bg\|^2 - Tr(B^TB) \leq -c_1ds/4\sqrt{m}\right]$.

Now noticing that $c_1ds/4\sqrt{m} \leq c_1(\sum_{i=1}^{m}\lambda_i(B^TB)^2)/\lambda_1(B^TB)$ by eq. (5), we may invoke the second relation in Lemma 9 on eq. (7) to get:

$$\begin{aligned} \mathbb{P}_g\left[\|Bg\|^2 - Tr(B^TB) \leq -c_1ds/4\sqrt{m}\right] \\ = \mathbb{P}_{\tilde{g}}\left[\sum_{i=1}^{d}\lambda_i(B^TB)(\tilde{g}_i^2 - 1) \leq -c_1ds/4\sqrt{m}\right] \geq c_1\exp\left(-C_1(c_1ds)^2/(16m\sum_{i=1}^{d}\lambda_i^2(B^TB))\right). \end{aligned}$$

Yielding that $6\delta^{1/8} \geq c_1\exp\left(-C_1(c_1ds)^2/(16m\sum_{i=1}^{d}\lambda_i^2(B^TB))\right)$.

Case 2: If $Tr(B^TB) \in [(1 - 2\varepsilon + c_1/(4\sqrt{m}))ds, \infty)$ then by eq. (6) (and the comment below the equation) we have with probability at least $1 - 6\delta^{1/8}$ over g that

$$\|Bg\|^2 - Tr(B^TB) \leq ((1 + 2\varepsilon + \varepsilon^2) - (1 - 2\varepsilon + c_1/(4\sqrt{m})))ds < 5\varepsilon ds.$$

implying that $6\delta^{1/8} \geq \mathbb{P}_g\left[\|Bg\|^2 - Tr(B^TB) \geq 5\varepsilon ds\right]$.

Now using the first relation in Lemma 9 combined with eq. (7), it follows that

$$\begin{aligned} \mathbb{P}_g\left[\|Bg\|^2 - Tr(B^TB) > 5\varepsilon ds\right] \\ = \mathbb{P}_{\tilde{g}}\left[\sum_{i=1}^{d}\lambda_i(B^TB)(\tilde{g}_i^2 - 1) > 5\varepsilon ds\right] \geq c_1\exp\left(-C_1(5\varepsilon ds)^2/(\sum_{i=1}^{d}\lambda_i^2(B^TB))\right). \end{aligned}$$

Yielding that $6\delta^{1/8} \geq c_1\exp\left(-C_1(5\varepsilon ds)^2/(\sum_{i=1}^{d}\lambda_i^2(B^TB))\right)$.

Conclusion. Now using that $m \geq \varepsilon^{-2}\lg(1/\delta)$ and $c_1 \leq 1$ it follows that $c_1^2/16m \leq 5^2\varepsilon^2$ which then implies that $C_1(c_1 ds)^2/(16m\sum_{i=1}^d \lambda_i^2(B^TB)) \leq C_1(5\varepsilon ds)^2/(\sum_{i=1}^d \lambda_i^2(B^TB)$. Combining this with the conclusion of the above two cases, we get that $6\delta^{1/8} \geq c_1 \exp\left(-C_1(5\varepsilon ds)^2/(\sum_{i=1}^d \lambda_i^2(B^TB))\right)$. With this relation, choosing the universal constant $C = (c_1/6)^{16}$ (less than $1/36$ as used in eq. (3)), which implies that $c_1/(6\delta^{1/16}) \geq 1$, and using that $m = t\varepsilon^{-2}\lg(1/\delta)$, we now get that

$$\lg(6\delta^{1/8}) \geq \lg(c_1) - C_1(5\varepsilon ds)^2/(\sum_{i=1}^d \lambda_i^2(B^TB))$$

$$\Rightarrow \sum_{i=1}^d \lambda_i^2(B^TB) \leq C_1(5\varepsilon ds)^2/(\lg(c_1/(6\delta^{1/8}))) \leq C_1 16(5\varepsilon ds)^2/\lg(1/\delta) \leq 20^2 C_1 t(ds)^2/m \tag{8}$$

We now define the vector $w \in \mathbb{R}^d$ as

$$[w]_i = \begin{cases} 1 & \text{if } \lambda_i(B^TB) \geq ds/(3m) \\ \\ 0 & \text{else} \end{cases}$$

and let l be equal to the number of non-zero entries of w. Let further λ denote the vector in $\mathbb{R}^d$ with the eigenvalues of B^TB in decreasing order. It then follows by Cauchy-Schwartz and eq. (8) that we have the following upper bound on the sum of the eigenvalues of B^TB larger than $ds/(3m)$:

$$\sum_{i:\lambda_i(B^TB)\geq ds/(3m)} \lambda_i(B^TB) = \langle \lambda, w\rangle \leq \|\lambda\|\,\|w\| = \sqrt{\sum_{i=1}^d \lambda_i^2(B^TB) l} \leq \sqrt{20^2 C_1 t(ds)^2 l/m}.$$

At the same time, we get the following lower bound on the sum of the eigenvalues of B^TB larger than $ds/(3m)$ by eq. (3) and the fact that (B^TB) has rank at most m and hence at most m non-zero eigenvalues

$$\sum_{i:\lambda_i(B^TB)\geq ds/(3m)} \lambda_i(B^TB) = \sum_{i=1}^d \lambda_i(B^TB) - \sum_{i:\lambda_i(B^TB)< ds/(3m)} \lambda_i(B^TB) \geq 2ds/3 - ds/3 = ds/3.$$

Hence combining the upper and lower bound we obtain that $ds/3 \leq \sqrt{20^2 C_1 t(ds)^2 l/m}$, implying that $m/(60^2 C_1 t) \leq l$, which by setting c in Lemma 8 equal to $1/60^2 C_1 \leq 1$ ($C_1 \geq 1$ by Lemma 9) concludes the proof of Lemma 8. □

References

[1] D. Achlioptas. Database-friendly random projections: Johnson-lindenstrauss with binary coins. *J. Comput. Syst. Sci.*, 66(4):671–687, 2003.

[2] N. Ailon and B. Chazelle. The fast johnson–lindenstrauss transform and approximate nearest neighbors. *SIAM J. Comput.*, 39:302–322, 2009.

[3] N. Ailon and E. Liberty. Fast dimension reduction using rademacher series on dual BCH codes. In S. Teng, editor, *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2008, San Francisco, California, USA, January 20-22, 2008*, pages 1–9. SIAM, 2008.

[4] N. Alon, R. Panigrahy, and S. Yekhanin. Deterministic approximation algorithms for the nearest codeword problem. In I. Dinur, K. Jansen, J. Naor, and J. D. P. Rolim, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, 12th International Workshop, APPROX 2009, and 13th International Workshop, RANDOM 2009, Berkeley, CA, USA, August 21-23, 2009. Proceedings*, volume 5687 of *Lecture Notes in Computer Science*, pages 339–351. Springer, 2009.

[5] R. I. Arriaga and S. S. Vempala. An algorithmic theory of learning: Robust concepts and random projection. *Mach. Learn.*, 63(2):161–182, 2006.

[6] S. Bamberger and F. Krahmer. Optimal fast johnson–lindenstrauss embeddings for large data sets. *Sampling Theory, Signal Processing, and Data Analysis*, 19(1):3, 2021.

[7] B. Chazelle. A spectral approach to lower bounds with applications to geometric searching. *SIAM Journal on Computing*, 27(2):545–556, 1998.

[8] T. T. Do, L. Gan, Y. Chen, N. Nguyen, and T. D. Tran. Fast and efficient dimensionality reduction using structurally random matrices. In *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1821–1824, 2009.

[9] Z. Dvir, A. Golovnev, and O. Weinstein. Static data structure lower bounds imply rigidity. In M. Charikar and E. Cohen, editors, *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019*, pages 967–978. ACM, 2019.

[10] O. N. Fandina, M. M. Høgsgaard, and K. G. Larsen. The fast johnson-lindenstrauss transform is even faster. *CoRR*, abs/2204.01800, 2022.

[11] C. Freksen, L. Kamma, and K. G. Larsen. Fully understanding the hashing trick. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 5394–5404, Red Hook, NY, USA, 2018. Curran Associates Inc.

[12] C. B. Freksen and K. G. Larsen. On using toeplitz and circulant matrices for johnson-lindenstrauss transforms. *Algorithmica*, 82(2):338–354, 2020.

[13] J. Friedman. A note on matrix rigidity. *Comb.*, 13(2):235–239, 1993.

[14] A. Hinrichs and J. Vybíral. Johnson-lindenstrauss lemma for circulant matrices**. *Random Structures & Algorithms*, 39(3):391–398, 2011.

[15] P. Indyk and R. Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing*, STOC '98, page 604–613, New York, NY, USA, 1998. Association for Computing Machinery.

[16] M. Jagadeesan. Understanding sparse JL for feature hashing. In H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 15177–15187, 2019.

[17] V. Jain, N. S. Pillai, and A. Smith. Kac meets johnson and lindenstrauss: a memory-optimal, fast johnson-lindenstrauss transform. *CoRR*, abs/2003.10069, 2020. To appear in Annals of Applied Probability.

[18] W. Johnson and J. Lindenstrauss. Extensions of lipschitz maps into a hilbert space. *Contemporary Mathematics*, 26:189–206, 01 1984.

[19] M. Kac. Foundations of kinetic theory. In *Proceedings of The third Berkeley symposium on mathematical statistics and probability*, pages 171–197. University of California Press Berkeley and Los Angeles, California, 1958.

[20] D. M. Kane, R. Meka, and J. Nelson. Almost optimal explicit johnson-lindenstrauss families. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques - 14th International Workshop, APPROX 2011, and 15th International Workshop, RANDOM 2011, Princeton, NJ, USA, August 17-19, 2011. Proceedings*, pages 628–639, 2011.

[21] D. M. Kane and J. Nelson. Sparser johnson-lindenstrauss transforms. *J. ACM*, 61(1):4:1–4:23, 2014.

[22] F. Krahmer and R. Ward. New and improved johnson-lindenstrauss embeddings via the restricted isometry property. *SIAM J. Math. Anal.*, 43(3):1269–1281, 2011.

[23] K. G. Larsen. Constructive discrepancy minimization with hereditary L2 guarantees. In R. Niedermeier and C. Paul, editors, *36th International Symposium on Theoretical Aspects of Computer Science, STACS 2019, March 13-16, 2019, Berlin, Germany*, volume 126 of *LIPIcs*, pages 48:1–48:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.

[24] K. G. Larsen and J. Nelson. The johnson-lindenstrauss lemma is optimal for linear dimensionality reduction. In *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy*, pages 82:1–82:11, 2016.

[25] K. G. Larsen and J. Nelson. Optimality of the johnson-lindenstrauss lemma. In C. Umans, editor, *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 633–638. IEEE Computer Society, 2017.

[26] J. Morgenstern. On linear algorithms. In Z. Kohavi and A. Paz, editors, *Theory of Machines and Computations*, pages 59–66. Academic Press, 1971.

[27] J. Morgenstern. Note on a lower bound on the linear complexity of the fast fourier transform. *J. ACM*, 20:305–306, 1973.

[28] J. Nelson and H. L. Nguyen. Sparsity lower bounds for dimensionality reducing maps. In D. Boneh, T. Roughgarden, and J. Feigenbaum, editors, *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 101–110. ACM, 2013.

[29] S. Saraf and S. Yekhanin. Noisy interpolation of sparse polynomials, and applications. In *Proceedings of the 26th Annual IEEE Conference on Computational Complexity, CCC 2011, San Jose, California, USA, June 8-10, 2011*, pages 86–92. IEEE Computer Society, 2011.

[30] L. G. Valiant. Graph-theoretic arguments in low-level complexity. In J. Gruska, editor, *Mathematical Foundations of Computer Science 1977, 6th Symposium, Tatranska Lomnica, Czechoslovakia, September 5-9, 1977, Proceedings*, volume 53 of *Lecture Notes in Computer Science*, pages 162–176. Springer, 1977.

[31] J. Vybiral. A variant of the johnson-lindenstrauss lemma for circulant matrices. *Journal of Functional Analysis*, 260:1096–1105, 02 2010.

[32] M. J. Wainwright. *Basic tail and concentration bounds*, page 21–57. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2019.

[33] K. Q. Weinberger, A. Dasgupta, J. Langford, A. J. Smola, and J. Attenberg. Feature hashing for large scale multitask learning. In *Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, June 14-18, 2009*, pages 1113–1120, 2009.

[34] A. R. Zhang and Y. Zhou. On the non-asymptotic and sharp lower tail bounds of random variables. *Stat*, 9(1):e314, 2020. e314 sta4.314.