OSNAP: Faster numerical linear algebra algorithms via sparser subspace embeddings

Jelani Nelson* Huy L. Nguyễn†

Abstract

An *oblivious subspace embedding (OSE)* given some parameters ε, d is a distribution $\mathcal{D}$ over matrices $\Pi \in \mathbb{R}^{m\times n}$ such that for any linear subspace $W \subseteq \mathbb{R}^n$ with $\dim(W) = d$ it holds that

$$\mathbb{P}_{\Pi\sim\mathcal{D}}(\forall x \in W \ \|\Pi x\|_2 \in (1\pm\varepsilon)\|x\|_2) > 2/3.$$

We show that the sparse Johnson-Lindenstrauss constructions of [Kane-Nelson, SODA 2012] provide OSE's with $m = O(d^{1+\gamma}/\varepsilon^2)$, and where every matrix Π in the support of the OSE has only $s = O_\gamma(1/\varepsilon)$ non-zero entries per column. The value $\gamma > 0$ can be any desired constant. Our m is nearly optimal since $m \geq d$ is required simply to ensure no non-zero vector of W lands in the kernel of Π. Our work gives the first OSE's with $m = o(d^2)$ to have $s = o(d)$. We also identify a certain a class of distributions, which we call Oblivious Sparse Norm-Approximating Projections (OSNAPs), such that any distribution in this class provides this guarantee.

Plugging OSNAPs into known algorithms for approximate ordinary least squares regression, ℓ_p regression, low rank approximation, and approximating leverage scores implies faster algorithms for all these problems. For example, for the approximate least squares regression problem of computing x that minimizes $\|Ax - b\|_2$ up to a constant factor, our embeddings imply a running time of $\tilde{O}(\text{nnz}(A) + r^\omega)$.[1] Here $r = \text{rank}(A)$, $\text{nnz}(\cdot)$ counts non-zero entries, and ω is the exponent of matrix multiplication. Previous algorithms had a worse dependence on r.

Our main result is essentially a Bai-Yin type theorem in random matrix theory and is likely to be of independent interest: we show that for any fixed $U \in \mathbb{R}^{n\times d}$ with orthonormal columns and random sparse Π, all singular values of ΠU lie in $[1-\varepsilon, 1+\varepsilon]$ with good probability. Our main result is accomplished via the classical moment method, i.e. by bounding $\mathbb{E}\text{tr}(((\Pi U)^*\Pi U - I)^\ell)$ for $\ell = \Theta(\log d)$. We also show that taking $\ell = 2$ allows one to recover a slightly sharper version of the main result of [Clarkson-Woodruff, STOC 2013] with considerably less effort. That is, we show that one obtains an OSE with $m = O(d^2/\varepsilon^2)$, $s = 1$. The quadratic dependence on d is optimal [Nelson-Nguyễn, STOC 2013].

1 Introduction

There has been much recent work on applications of dimensionality reduction to handling large datasets. Typically special features of the data such as low "intrinsic" dimensionality, or sparsity,

*Institute for Advanced Study. minilek@ias.edu. Supported by NSF CCF-0832797 and NSF DMS-1128155.

†Princeton University. hlnguyen@princeton.edu. Supported in part by NSF CCF-0832797 and a Gordon Wu fellowship.

[1] We say $g = \tilde{\Omega}(f)$ when $g = \Omega(f/\text{polylog}(f))$, $g = \tilde{O}(f)$ when $g = O(f\cdot\text{polylog}(f))$, and $g = \tilde{\Theta}(f)$ when $g = \tilde{\Omega}(f)$ and $g = \tilde{O}(f)$ simultaneously.

are exploited to reduce the volume of data before processing, thus speeding up analysis time. One success story of this approach is the applications of fast algorithms for the Johnson-Lindenstrauss (JL) lemma [24], which allows one to reduce the dimensionality of a set of vectors while preserving all pairwise distances. There have been two popular lines of work in this area: one focusing on fast embeddings for all vectors [2–4, 23, 31, 32, 47], and one focusing on fast embeddings specifically for sparse vectors [1, 7, 15, 25, 26].

In this work we focus on the problem of constructing an *oblivious subspace embedding (OSE)* [40] and on applications of these embeddings. Roughly speaking, the problem is to design a data-independent distribution over linear mappings such that when data come from an *unknown* low-dimensional subspace, they are reduced to roughly their true dimension while their structure (all distances in the subspace in this case) is preserved at the same time. It can be seen as a continuation of the approach based on the JL lemma to subspaces, and these embeddings have found applications in numerical linear algebra problems such as least squares regression, ℓ_p regression, low rank approximation, and approximating leverage scores [11–13, 17, 38, 40, 44]. We refer the interested reader to the surveys [20, 33] for an overview. Here we focus on the setting of sparse inputs, where it is important that the algorithms take time proportional to the input sparsity.

Throughout this document we use $\|\cdot\|$ to denote ℓ_2 norm in the case of vector arguments, and $\ell_{2\to2}$ operator norm in the case of matrix arguments. Recall the definition of the OSE problem.

Definition 1. *An* oblivious subspace embedding (OSE) *is a distribution over $m \times n$ matrices Π such that for any d-dimensional subspace $W \subset \mathbb{R}^n$, $\mathbb{P}_{\Pi\sim\mathcal{D}}(\forall x \in W\ \|\Pi x\|_2 \in (1\pm\varepsilon)\|x\|_2) > 2/3$. Here n, d, ε are given parameters of the problem and we would like m as small as possible.*

OSE's were first introduced in [40] as a means to obtain fast randomized algorithms for several numerical linear algebra problems. To see the connection, consider for example the least squares regression problem of computing $\operatorname{argmin}_{x\in\mathbb{R}^d}\|Ax-b\|$ for some $A \in \mathbb{R}^{n\times d}$. Suppose $\Pi \in \mathbb{R}^{m\times n}$ preserves the ℓ_2 norm up to $1\pm\varepsilon$ of all vectors in the subspace spanned by b and the columns of A. Let $\tilde{x} = \operatorname{argmin}_x\|\Pi Ax - \Pi b\|$ and $x^* = \operatorname{argmin}_x\|Ax-b\|$. Then

$$(1-\varepsilon)\|A\tilde{x}-b\| \le \|\Pi A\tilde{x}-\Pi b\| \le \|\Pi Ax^*-\Pi b\| \le (1+\varepsilon)\|Ax^*-b\|.$$

Thus $\tilde{x}$ provides a solution within $(1+\varepsilon)/(1-\varepsilon) = 1+O(\varepsilon)$ of optimal. Since this subspace has dimension at most $d+1$, one only needs m being some function of ε, d. Thus the running time for approximate $n\times d$ regression becomes that for $m\times d$ regression, plus an additive term for the time required to compute $\Pi A, \Pi b$. This is a gain for instances with $n \gg d$. Also, the $2/3$ success probability guaranteed by Definition 1 can be amplified to $1-\delta$ by running this procedure $O(\log(1/\delta))$ times with independent randomness and taking the best $\tilde{x}$ found in any run. We furthermore point out that another reduction from $(1+\varepsilon)$-approximate least squares regression to OSE's via preconditioning followed by gradient descent actually only needs an OSE with constant distortion independent of ε (see [13]), so that $\varepsilon = \Theta(1)$ in an OSE is of primary interest.

It is known that a random matrix with independent subgaussian entries and $m = O(d/\varepsilon^2)$ provides an OSE with $1+\varepsilon$ distortion [19, 30] . Unfortunately, the time to compute ΠA is then larger than the known $\tilde{O}(nd^{\omega-1})$ time bound to solve the exact regression problem itself, where $\omega < 2.373\ldots$ [49] is the exponent of square matrix multiplication. In fact, since $m \ge d$ in any OSE, dividing Π, A into $d\times d$ blocks and using fast square matrix multiplication to then multiply ΠA would yield time $\Theta(mnd^{\omega-2})$, which is at least $\Omega(nd^{\omega-1})$. Thus implementing the approach of the previous paragraph naively provides no gains. The work of [40] overcame this barrier by

choosing a special Π so that ΠA can be computed in time $O(nd \log n)$ (see also [44]). This matrix Π was the Fast JL Transform of [2], which has the property that Πx can be computed in roughly $O(n \log n)$ time for any $x \in \mathbb{R}^n$. Thus, multiplying ΠA by iterating over columns of A gives the desired speedup.

The $O(nd \log n)$ running time of the above scheme to compute ΠA seems almost linear, and thus nearly optimal, since the input size to describe A is nd. While this is true for dense A, in many applications one expects A to be sparse, in which case linear in the input description actually means $O(\text{nnz}(A))$, where $\text{nnz}(\cdot)$ counts non-zero entries. For example, one numerical linear algebra problem of wide interest is matrix completion, where one assumes that some small number of entries in a low rank matrix A have been revealed, and the goal is to then recover A. This problem arises in recommendation systems, where for example the rows of A represent users and the columns represent products, and $A_{i,j}$ is the rating of product j by customer i. One wants to infer "hidden ratings" to then make product recommendations, based on the few ratings that customers have actually made. Such matrices are usually very sparse; when for example $A_{i,j}$ is user i's score for movie j in the Netflix matrix, only roughly 1% of the entries of A are known [51]. Some matrix completion algorithms work by iteratively computing singular value decompositions (SVDs) of various matrices that have the same sparsity as the initial A, then thresholding the result to only contain the large singular values then re-sparsifying [9]. Furthermore it was empirically observed that the matrix iterates were low rank, so that a fast low rank approximation algorithm for sparse matrices, as what is provided in this work, could replace full SVD computation to give speedup.

In a recent beautiful and surprising work, [13] showed that there exist OSE's with $m = \text{poly}(d/\varepsilon)$, and where every matrix Π in the support of the distribution is *very* sparse: even with only $s = 1$ non-zero entry per column! Thus one can transform, for example, an $n \times d$ least squares regression problem into a $\text{poly}(\text{d}/\varepsilon) \times d$ regression problem by multiplying ΠA in $\text{nnz}(A) \cdot s = \text{nnz}(A)$ time. The work [13] gave two sparse OSE's: one with $m = O(d^2 \log^6(d/\varepsilon)/\varepsilon^2), s = 1$, and another with $m = \tilde{O}(d^2 \log(1/\delta)/\varepsilon^4 + d \log^2(1/\delta)/\varepsilon^4), s = O(\log(d/\delta)/\varepsilon)$. The second construction has the benefit of providing a subspace embedding with success probability $1 - \delta$ and not just $2/3$, which is important e.g. in known reductions from ℓ_p regression to OSE's [11].

Our Main Contribution: We give OSE's with $m = O(d^{1+\gamma}/\varepsilon^2), s = O_\gamma(1/\varepsilon)$, where $\gamma > 0$ can be any constant. Note s does not depend on d. The constant hidden in the O_γ is $\text{poly}(1/\gamma)$. The success probability is $1 - 1/d^c$ for any desired constant c. One can also set $m = O(d \cdot \text{polylog}(d/\delta)/\varepsilon^2), s = \text{polylog}(d/\delta)/\varepsilon$ for success probability $1 - \delta$. Ours are the first analyses to give OSE's having $m = o(d^2)$ with $s = o(d)$. Observe that in both our parameter settings m is nearly linear in d, which is nearly optimal since any OSE must have $m \geq d$ simply to ensure no non-zero vector of the subspace is in the kernel of Π. We also show that a simpler instantiation of our approach gives $m = O(d^2/\varepsilon^2), s = 1$, recovering a sharpening of the main result of [13] with a much simpler proof. Our quadratic dependence on d is optimal for $s = 1$ [37].

Plugging our improved OSE's into previous work implies faster algorithms for several numerical linear algebra problems, such as approximate least squares regression, ℓ_p regression, low rank approximation, and approximating leverage scores. In fact for all these problems, except approximating leverage scores, known algorithms only make use of OSE's with distortion $\Theta(1)$ independent of the desired $1+\varepsilon$ approximation guarantee, in which case our matrices have $m = O(d^{1+\gamma}), s = O_\gamma(1)$, i.e. constant column sparsity and a near-optimal number of rows.

reference	regression	leverage scores	low rank approximation
[13]	$O(\text{nnz}(A)) + \tilde{O}(d^3)$ $\tilde{O}(\text{nnz}(A) + r^3)$	$\tilde{O}(\text{nnz}(A) + r^3)$	$O(\text{nnz}(A)) + \tilde{O}(nk^2)$
this work	$O_\gamma(\text{nnz}(A)) + O(d^{\omega+\gamma})$ $\tilde{O}(\text{nnz}(A) + r^\omega)$	$\tilde{O}(\text{nnz}(A) + r^\omega)$	$O_\gamma(\text{nnz}(A)) + \tilde{O}(nk^{\omega-1+\gamma} + k^{\omega+\gamma})$

Figure 1: The improvement gained in running times by using our OSE's, where $\gamma > 0$ is an arbitrary constant. Dependence on ε suppressed for readability; see Section 3 for dependence.

We also remark that the analyses of [13] require $\Omega(d)$-wise independent hash functions, so that from the seed used to generate Π naively one needs an additive $\Omega(d)$ time to identify the non-zero entries in each column just to evaluate the hash function. In streaming applications this can be improved to additive $\tilde{O}(\log^2 d)$ time using fast multipoint evaluation of polynomials (see [27, Remark 16]), though ideally if $s = 1$ one could hope for a construction that allows one to find, for any column, the non-zero entry in that column in constant time given only a short seed that specifies Π (i.e. without writing down Π explicitly in memory, which could be prohibitively expensive for n large in applications such as streaming and out-of-core numerical linear algebra). Recall that in the entry-wise turnstile streaming model, A receives entry-wise updates of the form $((i, j), v)$, which cause the change $A_{i,j} \leftarrow A_{i,j} + v$. Updating the embedding thus amounts to adding v times the jth row of Π to ΠA, which should ideally take $O(s)$ time and not $O(s) + \tilde{O}(\log^2 d)$. Our analyses only use 4-wise independent hash functions when $s = 1$ and $O(\log d)$-wise independent hash functions for larger s, thus allowing fast computation of any column of Π from a short seed.

1.1 Problem Statements and Bounds

We now formally define all numerical linear algebra problems we consider. Plugging our new OSE's into previous algorithms provides speedup for all these problems (see Figure 1; the consequences for ℓ_p regression are also given in Section 3). The value r used in bounds denotes $\text{rank}(A)$. In what follows, $b \in \mathbb{R}^n$ and $A \in \mathbb{R}^{n\times d}$.

Leverage Scores: Let $A = U\Sigma V^*$ be the SVD. Output the row ℓ_2 norms of U up to $1 \pm \varepsilon$.

Least Squares Regression: Compute $\tilde{x} \in \mathbb{R}^d$ so that $\|A\tilde{x} - b\| \leq (1 + \varepsilon) \cdot \min_{x\in\mathbb{R}^d} \|Ax - b\|$.

ℓ_p Regression ($p \in [1, \infty)$): Compute $\tilde{x} \in \mathbb{R}^d$ so that $\|A\tilde{x} - b\|_p \leq (1 + \varepsilon) \cdot \min_{x\in\mathbb{R}^d} \|Ax - b\|_p$.

Low Rank Approximation: Given integer $k > 0$, compute $\tilde{A}_k \in \mathbb{R}^{n\times d}$ with $\text{rank}(\tilde{A}) \leq k$ so that $\|A - \tilde{A}_k\|_F \leq (1 + \varepsilon) \cdot \min_{\text{rank}(A_k)\leq k} \|A - A_k\|_F$, where $\|\cdot\|_F$ is Frobenius norm.

1.2 Our Approach

Let $\Pi \in \mathbb{R}^{m\times n}$ be a sparse JL matrix as in [26]. For example, one such construction is to choose each column of Π independently, and within a column we pick exactly s random locations (without replacement) and set the corresponding entries to $\pm 1/\sqrt{s}$ at random with all other entries in the column then set to zero. Observe any d-dimensional subspace $W \subseteq \mathbb{R}^n$ satisfies $W = \{x : \exists y \in \mathbb{R}^d, x = Uy\}$ for some $U \in \mathbb{R}^{n\times d}$ whose columns form an orthonormal basis for W. A matrix Π preserving ℓ_2 norms of all $x \in W$ up to $1\pm\varepsilon$ is thus equivalent to the statement $\|\Pi Uy\| = (1\pm\varepsilon)\|Uy\|$

simultaneously for all $y \in \mathbb{R}^d$. This is equivalent to $\|\Pi U y\| = (1 \pm \varepsilon)\|y\|$ since $\|Uy\| = \|y\|$. This in turn is equivalent to all singular values of ΠU lying in the interval $[1-\varepsilon, 1+\varepsilon]$.[2] Write $S = (\Pi U)^* \Pi U$, so that we want to show all eigenvalues of S lie in $[(1-\varepsilon)^2, (1+\varepsilon)^2]$. That is, we want to show

$$(1-\varepsilon)^2 \leq \inf_{\|y\|=1} \|Sy\| \leq \sup_{\|y\|=1} \|Sy\| \leq (1+\varepsilon)^2.$$

By the triangle inequality we have $\|Sy\| = \|y\| \pm \|(S-I)y\|$. Thus, it suffices to show $\|S - I\| \leq \min\{1-(1-\varepsilon)^2, (1+\varepsilon)^2 - 1\} = 2\varepsilon - \varepsilon^2$ with good probability. By Markov's inequality

$$\mathbb{P}(\|S-I\| > t) = \mathbb{P}(\|S-I\|^\ell > t^\ell) < t^{-\ell} \cdot \mathbb{E}\|S-I\|^\ell \leq t^{-\ell} \cdot \mathbb{E}\mathrm{tr}((S-I)^\ell) \tag{1}$$

for any even integer $\ell \geq 2$. This is because if the eigenvalues of $S - I$ are $\lambda_1, \ldots, \lambda_d$, then those of $(S-I)^\ell$ are $\lambda_1^\ell, \ldots, \lambda_d^\ell$. Thus $\mathrm{tr}((S-I)^\ell) = \sum_i \lambda_i^\ell \geq \max_i |\lambda_i|^\ell = \|S-I\|^\ell$, since ℓ is even so that the λ_i^ℓ are nonnegative. Setting $\ell = 2$ allows $m = O(d^2/\varepsilon^2), s = 1$ with a simple proof (Theorem 4), and $\ell = \Theta(\log d)$ yields the main result with $s > 1$ and $m \approx d/\varepsilon^2$ (Theorem 10 and Theorem 13).

We remark that this method of bounding the range of singular values of a random matrix by computing the expectation of traces of large powers is a classical approach in random matrix theory (see the work of Bai and Yin [6]). Such bounds were also used in bounding operator norms of random matrices in work of Füredi and Komlós [18], and in computing the limiting spectral distribution by Wigner [48]. See also the surveys [42, 46]. We also remark that this work can be seen as the natural non-commutative extension of the work on the sparse JL lemma itself. Indeed, if one imagines that $d = 1$ so that $U = u \in \mathbb{R}^{n \times 1}$ is a "1-dimensional matrix" with orthonormal columns (i.e. a unit vector), then preserving the 1-dimensional subspace spanned by u with probability $1 - \delta$ is equivalent to preserving the ℓ_2 norm of u with probability $1 - \delta$. Indeed, in this case the expression $\|S - I\|$ in Eq. (1) is simply $|\|\Pi u\|^2 - 1|$. This is *exactly* the JL lemma, where one achieves $m = O(1/(\varepsilon^2\delta)), s = 1$ by a computation of the second moment [43], and $m = O(\log(1/\delta)/\varepsilon^2), s = O(\log(1/\delta)/\varepsilon)$ by a computation of the $O(\log(1/\delta))$th moment [26].

Our approach is very different from that of Clarkson and Woodruff [13]. For example, take the $s = 1$ construction so that Π is specified by a random hash function $h : [n] \to [m]$ and a random $\sigma \in \{-1, 1\}^n$, where $[n] \stackrel{\text{def}}{=} \{1, \ldots, n\}$. For each $i \in [n]$ we set $\Pi_{h(i),i} = \sigma_i$, and every other entry in Π is set to zero. The analysis in [13] then worked roughly as follows: let $\mathcal{I} \subset [n]$ denote the set of "heavy" rows, i.e. those rows u_i of U where $\|u_i\|$ is "large". We write $x = x_{\mathcal{I}} + x_{[n]\setminus\mathcal{I}}$, where x_S for a set S denotes x with all coordinates in $[n]\setminus S$ zeroed out. Then $\|x\|^2 = \|x_{\mathcal{I}}\|^2 + \|x_{[n]\setminus\mathcal{I}}\|^2 + 2\langle x_{\mathcal{I}}, x_{[n]\setminus\mathcal{I}}\rangle$. The argument in [13] conditioned on $\mathcal{I}$ being perfectly hashed by h so that $\|x_{\mathcal{I}}\|^2$ is preserved exactly. Using an approach in [25, 26] based on the Hanson-Wright inequality [21] together with a net argument, [13] argued that $\|x_{[n]\setminus\mathcal{I}}\|^2$ is preserved simultaneously for all $x \in W$; this step required $\Omega(d)$-wise independence to union bound over the net. A simpler concentration argument handled $\langle x_{\mathcal{I}}, x_{[n]\setminus\mathcal{I}}\rangle$. This type of analysis led to $m = \tilde{O}(d^4/\varepsilon^4), s = 1$. A slightly more involved refinement of the analysis, where one partitions the rows of U into multiple levels of "heaviness", led to the bound $m = O(d^2 \log^6(d/\varepsilon)/\varepsilon^2), s = 1$. The construction in [13] with similar m and larger s for $1 - \delta$ success probability followed a similar but more complicated analysis; that construction hashed $[n]$ into buckets then used the sparse JL matrices of [26] in each bucket. Meanwhile, our analyses use the matrices of [26] directly without the extra hashing step.

We remark that in our analyses, the properties we need from an OSE are the following.

[2] Recall that the singular values of a (possibly rectangular) matrix B are the square roots of the eigenvalues of B^*B, where $(\cdot)^*$ denotes conjugate transpose.

- For each Π in the support of the distribution, we can write $\Pi_{i,j} = \delta_{i,j}\sigma_{i,j}/\sqrt{s}$, where the σ are i.i.d. ± 1 random variables, and $\delta_{i,j}$ is an indicator random variable for the event $\Pi_{i,j} \neq 0$.
- $\forall\, j \in [n]$, $\sum_{i=1}^{m} \delta_{i,j} = s$ with probability 1, i.e. every column has *exactly* s non-zero entries.
- For any $S \subseteq [m] \times [n]$, $\mathbb{E} \prod_{(i,j)\in S} \delta_{i,j} \leq (s/m)^{|S|}$.
- The columns of Π are i.i.d.

We call any Π drawn from an OSE with the above properties an *oblivious sparse norm-approximating projection* (OSNAP). In our analyses, the last condition and the independence of the $\sigma_{i,j}$ can actually be weakened to only be (2ℓ)-wise independent, since we only use ℓth moment bounds.

We now sketch a brief technical overview of our proofs. When $\ell = 2$, we have $\mathrm{tr}((S-I)^2) = \|S-I\|_F^2$, and our analysis becomes a half-page computation (Theorem 4). For larger ℓ, we expand $\mathrm{tr}((S-I)^\ell)$ and compute its expectation. This expression is a sum of exponentially many monomials, each involving a product of ℓ terms. Without delving into all technical details, each such monomial can be thought of as being in correspondence with some undirected multigraph (see the dot product multigraphs in the proof of Theorem 10). We group monomials with isomorphic graphs, bound the contribution from each graph separately, then sum over all graphs. Multigraphs whose edges all have even multiplicity turn out to be easier to handle (Lemma 11). However most graphs G do not have this property. Informally speaking, the contribution of a graph turns out to be related to the product over its edges of the contribution of that edge. Let us informally call this "contribution" $F(G)$. Thus if $E' \subset E$ is a subset of the edges of G, we can write $F(G) \leq F((G|_{E'})^2)/2 + F((G|_{E\setminus E'})^2)/2$ by AM-GM, where squaring a multigraph means duplicating every edge, and $G|_{E'}$ is G with all edges in $E\setminus E'$ removed. This reduces back to the case of even edge multiplicities, but unfortunately the bound we desire on $F(G)$ depends exponentially on the number of connected components. Thus this step is bad, since if G is connected, then one of $G|_{E'}, G|_{E\setminus E'}$ can have *many* connected components for any choice of E'. For example if G is a cycle on N vertices, then any partition of the edges into two sets $E', E\setminus E'$ will have that either $G_{E'}$ or $G_{E\setminus E'}$ has at least $N/2$ components. We overcome this by showing that any $F(G)$ is bounded by some $F(G')$ with the property that every component of G' has two edge-disjoint spanning trees. We then put one such spanning tree into E' for each component, so that $G|_{E\setminus E'}$ and $G|_{E'}$ both have the same number of components as G.

Remark 2. The work [26] provided two approaches to handle large ℓ in the case of sparse JL. The first was much simpler and relied on the Hanson-Wright inequality [21]. The Hanson-Wright inequality does have a non-commutative generalization (see [39, Theorem 6.22]) which can handle $d > 1$. Unfortunately, one should recall that the proof using [21] in [26] required conditioning on the columns of Π forming a good code, specifically meaning that no two columns have their non-zero entries in more than $O(s^2/m)$ of the same rows. Such an analysis can be carried out in our current context, but like in [26], this good event occurring with positive probability requires $s^2/m = \Omega(\log n)$, i.e. $s = \Omega(\sqrt{m \log n})$. Since here $m = \Omega(d/\varepsilon^2)$, this means $s = \Omega(\sqrt{d \log n}/\varepsilon)$. This was acceptable in [26] since then d was 1, but it is too weak in our current context.

The second approach of [26] was graph-theoretic as in the current work, although the current context presents considerable complications. In particular, at some point in our analysis we must bound a summation of products of dot products of rows of U (see Eq. (4)). In the case $d = 1$, a row of $U = u$ is simply a scalar. In the case of two "rows" u_i, u_j actually being scalars, the bound $|\langle u_i, u_j\rangle| \leq \|u_i\| \cdot \|u_j\|$ is actually equality. The sparse JL work implicitly exploited this fact.

Meanwhile in our case, using this bound turns out to lead to no result at all, and we must make use of the fact that the columns of U are orthogonal to make progress.

Remark 3. There has been much previous work on the eigenvalue spectrum of sparse random matrices (e.g. [28, 29, 50]). However as far as we are aware, these works were only concerned with $U = I$ and $n = d$, and furthermore they were interested in bounding the largest singular value of Π, or the bulk eigenvalue distribution, whereas we want that *all* singular values are $1 \pm \varepsilon$. On the other hand many of these works did consider entries of Π coming from distributions more general than ± 1, although this is not relevant for our algorithmic purposes. Our biggest technical contribution comes from dealing with U not being the identity matrix, since in the case when $U = I$, all graphs G in our technical overview have no edges other than self-loops and are much simpler to analyze.

1.3 Other Related Work

Simultaneously and independently of this work, Mahoney and Meng [34] showed that one can set $m = O(d^4/\varepsilon^4), s = 1$. Their argument was somewhat similar, although rather than using $\|S - I\|^2 \leq \mathrm{tr}((S - I)^2)$ as in Eq. (1), [34] used the Gershgorin circle theorem. After receiving our manuscript as well as an independent tip from Petros Drineas, the authors produced a second version of their manuscript with a proof and result that match our Theorem 4. Their work also gives an alternate algorithm for $(1 + \varepsilon)$ approximate ℓ_p regression in $O(\mathrm{nnz}(A) \log n + \mathrm{poly}(d/\varepsilon))$ time, without using the reduction in [11]. Their ℓ_p regression algorithm has the advantage over this work and over [13] of requiring only $\mathrm{poly}(d)$ space, but has the disadvantage of only working for $1 \leq p \leq 2$, whereas both this work and [13] handle all $p \in [1, \infty)$.

Another simultaneous and independent related work is that of Miller and Peng [35]. They provide a subspace embedding with $m = (d^{1+\gamma} + \mathrm{nnz}(A)/d^3)/\varepsilon^2$, $s = 1$. Their embedding is non-oblivious, meaning the construction of Π requires looking at the matrix A. Their work has the advantage of smaller s by a factor $O_\gamma(1/\varepsilon)$ (although for all problems considered here except approximating leverage scores, one only needs OSE's with $\varepsilon = \Theta(1)$), and has the disadvantage of m depending on $\mathrm{nnz}(A) \geq n$, and being non-oblivious, so that they cannot provide a $\mathrm{poly}(d)$-space algorithm in one-pass streaming applications. Furthermore their embeddings do not have a property required for known applications of OSE's to the low rank approximation problem (namely the approximate matrix multiplication property of Eq. (18)), and it is thus not known how to use their embeddings for this problem. Their embeddings also only fit into the reduction from ℓ_p regression [11] when n, d are polynomially related due to their failure probability being $\Omega(1/\mathrm{poly}(d))$. For this regime they provide the same asymptotic speedup over [13] as our work.

2 Analysis

In this section let the orthonormal columns of $U \in \mathbb{R}^{n \times d}$ be denoted $u^1, \ldots, u^d$. We will implement Eq. (1) and show $\mathbb{E}\mathrm{tr}((S-I)^\ell) < t^\ell \cdot \delta$ for $t = 2\varepsilon - \varepsilon^2$ and $\delta \in (0, 1)$ a failure probability parameter. Before proceeding with our proofs below, observe that for all k, k'

$$\begin{aligned} S_{k,k'} &= \frac{1}{s} \sum_{r=1}^{m} \left(\sum_{i=1}^{n} \delta_{r,i} \sigma_{r,i} u_i^k \right) \left(\sum_{i=1}^{n} \delta_{r,i} \sigma_{r,i} u_i^{k'} \right) \\ &= \frac{1}{s} \sum_{i=1}^{n} u_i^k u_i^{k'} \cdot \left(\sum_{r=1}^{m} \delta_{r,i} \right) + \frac{1}{s} \sum_{r=1}^{m} \sum_{i \neq j} \delta_{r,i} \delta_{r,j} \sigma_{r,i} \sigma_{r,j} u_i^k u_j^{k'} \end{aligned}$$

$$= \langle u^k, u^{k'} \rangle + \frac{1}{s} \sum_{r=1}^{m} \sum_{i \neq j} \delta_{r,i} \delta_{r,j} \sigma_{r,i} \sigma_{r,j} u_i^k u_j^{k'}$$

Noting $\langle u^k, u^k \rangle = \|u^k\|^2 = 1$ and $\langle u^k, u^{k'} \rangle = 0$ for $k \neq k'$, we have for all k, k'

$$(S - I)_{k,k'} = \frac{1}{s} \sum_{r=1}^{m} \sum_{i \neq j} \delta_{r,i} \delta_{r,j} \sigma_{r,i} \sigma_{r,j} u_i^k u_j^{k'}. \tag{2}$$

2.1 Analysis for $\ell = 2$

We first show that one can set $m = O(d^2/\varepsilon^2), s = 1$ by performing a 2nd moment computation.

Theorem 4. *For Π an OSNAP with $s = 1$ and $\varepsilon \in (0, 1)$, with probability at least $1 - \delta$ all singular values of ΠU are $1 \pm \varepsilon$ as long as $m \geq \delta^{-1}(d^2 + d)/(2\varepsilon - \varepsilon^2)^2$, σ is 4-wise independent, and h is pairwise independent.*

Proof. We need only show $\mathbb{E}\mathrm{tr}((S - I)^2) \leq (2\varepsilon - \varepsilon^2)^2 \cdot \delta$. Since $\mathrm{tr}((S - I)^2) = \|S - I\|_F^2$, we bound the expectation of this latter quantity. We first deal with the diagonal terms of $S - I$. By Eq. (2),

$$\mathbb{E}(S - I)_{k,k}^2 = \sum_{r=1}^{m} \sum_{i \neq j} \frac{2}{m^2} (u_i^k)^2 (u_j^k)^2 \leq \frac{2}{m} \cdot \|u^k\|^4 = \frac{2}{m}.$$

Thus the diagonal terms in total contribute at most $2d/m$ to $\mathbb{E}\|S - I\|_F^2$.

We now focus on the off-diagonal terms. By Eq. (2), $\mathbb{E}(S - I)_{k,k'}^2$ is equal to

$$\frac{1}{m^2} \sum_{r=1}^{m} \sum_{i \neq j} \left((u_i^k)^2 (u_j^{k'})^2 + u_i^k u_i^{k'} u_j^k u_j^{k'} \right) = \frac{1}{m} \sum_{i \neq j} \left((u_i^k)^2 (u_j^{k'})^2 + u_i^k u_i^{k'} u_j^k u_j^{k'} \right).$$

Noting $0 = \langle u^k, u^{k'} \rangle^2 = \sum_{k=1}^{n} (u_i^k)^2 (u_i^{k'})^2 + \sum_{i \neq j} u_i^k u_i^{k'} u_j^k u_j^{k'}$ we have that $\sum_{i \neq j} u_i^k u_i^{k'} u_j^k u_j^{k'} \leq 0$, so

$$\mathbb{E}(S - I)_{k,k'}^2 \leq \frac{1}{m} \sum_{i \neq j} (u_i^k)^2 (u_j^{k'})^2 \leq \frac{1}{m} \|u^k\|^2 \cdot \|u^{k'}\|^2 = \frac{1}{m},$$

Summing over $k \neq k'$, the total contribution from off-diagonal terms to $\mathbb{E}\|S - I\|_F^2$ is at most $d(d - 1)/m$. Thus $\mathbb{E}\|S - I\|_F^2 \leq d(d + 1)/m$, so it suffices to set $m \geq \delta^{-1} d(d + 1)/(2\varepsilon - \varepsilon^2)^2$. ■

2.2 Analysis for $\ell = \Theta(\log d)$

We now show that one can set $m \approx d/\varepsilon^2$, for slightly larger s by performing a $\Theta(\log d)$th moment computation. Before proceeding, it is helpful to state a few facts that we will repeatedly use. Recall that u^i denotes the ith column of U, and we will let u_i denote the ith row of U.

Lemma 5. $\sum_{k=1}^{n} u_k u_k^* = I$.

Proof.

$$\left(\sum_{k=1}^{n} u_k u_k^* \right)_{i,j} = e_i^* \left(\sum_{k=1}^{n} u_k u_k^* \right) e_j = \sum_{k=1}^{n} (u_k)_i (u_k)_j = \langle u^i, u^j \rangle,$$

and this inner product is 1 for $i = j$ and 0 otherwise. ■

Lemma 6. *For all $i \in [n]$, $\|u_i\| \le 1$.*

Proof. We can extend U to some orthogonal matrix $U' \in \mathbb{R}^{n\times n}$ by appending $n-d$ columns. For the rows u_i' of U' we then have $\|u_i\| \le \|u_i'\| = 1$. ■

Theorem 7 ([36, 45])**.** *A multigraph G has k edge-disjoint spanning trees iff*

$$|E_P(G)| \ge k(|P|-1)$$

for every partition P of the vertex set of G, where $E_P(G)$ is the set of edges of G crossing between two different partitions in P.

The following corollary is standard, and we will later only need it for the case $k = 2$.

Corollary 8. *Let G be a multigraph formed by removing at most k edges from a multigraph G' that has edge-connectivity at least $2k$. Then G must have at least k edge-disjoint spanning trees.*

Proof. For any partition P of the vertex set, each partition must have at least $2k$ edges leaving it in G'. Thus the number of edges crossing partitions must be at least $k|P|$ in G', and thus at least $k|P| - k$ in G. Theorem 7 thus implies that G has k edge-disjoint spanning trees. ■

Fact 9. *For any matrix $B \in \mathbb{C}^{d\times d}$, $\|B\| = \sup_{\|x\|,\|y\|=1} x^*By$.*

Proof. We have $\sup_{\|x\|,\|y\|=1} x^*By \le \|B\|$ since $x^*By \le \|x\| \cdot \|B\| \cdot \|y\|$. To show that unit norm x, y exist which achieve $\|B\|$, let $B = U\Sigma V^*$ be the singular value decomposition of B. That is, U, V are unitary and Σ is diagonal with entries $\sigma_1 \ge \sigma_2 \ge \ldots \sigma_d \ge 0$ so that $\|B\| = \sigma_1$. We can then achieve $x^*By = \sigma_1$ by letting x be the first column of U and y be the first column of V. ■

Theorem 10. *For Π an OSNAP with $s = \Theta(\log^3(d/\delta)/\varepsilon)$ and $\varepsilon \in (0,1)$, with probability at least $1-\delta$, all singular values of ΠU are $1 \pm \varepsilon$ as long as $m = \Omega(d\log^6(d/\delta)/\varepsilon^2)$ and σ, h are $\Omega(\log(d/\delta))$-wise independent.*

Proof. We bound $\mathbb{E}\mathrm{tr}((S-I)^\ell)$ for $\ell = \Theta(\log d)$ an even integer then apply Eq. (1). By induction on ℓ, for any $B \in \mathbb{R}^{n\times n}$ and $\ell \ge 1$,

$$(B^\ell)_{i,j} = \sum_{\substack{t_1,\ldots,t_{\ell+1}\in[n]\\ t_1=i, t_{\ell+1}=j}} \prod_{k=1}^{\ell} B_{t_k,t_{k+1}}, \text{ and thus } \mathrm{tr}(B^\ell) = \sum_{\substack{t_1,\ldots,t_{\ell+1}\in[n]\\ t_1=t_{\ell+1}}} \prod_{k=1}^{\ell} B_{t_k,t_{k+1}}.$$

Applying this identity to $B = S - I$ yields

$$\mathbb{E}\mathrm{tr}((S-I)^\ell) = \frac{1}{s^\ell}\cdot \mathbb{E} \sum_{\substack{k_1,k_2,\ldots,k_{\ell+1}\\ k_1=k_{\ell+1}\\ i_1\ne j_1,\ldots,i_\ell\ne j_\ell\\ r_1,\ldots,r_\ell}} \prod_{t=1}^{\ell} \delta_{r_t,i_t}\delta_{r_t,j_t}\sigma_{r_t,i_t}\sigma_{r_t,j_t}u_{i_t}^{k_t}u_{j_t}^{k_{t+1}}. \tag{3}$$

We now outline the strategy to bound Eq. (3). For each monomial ψ appearing on the right hand side of Eq. (3) we associate a three-layered undirected multigraph G_ψ with labeled edges and unlabeled vertices. We call these three layers the *left*, *middle*, and *right* layers, and we refer

to vertices in the left layer as *left vertices*, and similarly for vertices in the other layers. Define $y = |\{i_1, \dots, i_\ell, j_1, \dots, j_\ell\}|$ and $z = |\{r_1, \dots, r_\ell\}|$. The graph G_ψ has ℓ left vertices, y middle vertices corresponding to the distinct i_t, j_t in ψ, and z right vertices corresponding to the distinct r_t. For the sake of brevity, often we refer to the vertex corresponding to i_t (resp. j_t, r_t) as simply i_t (resp. j_t, r_t). Thus note that when we refer to for example some vertex i_t, it may happen that some other $i_{t'}$ or $j_{t'}$ is also the same vertex. We now describe the edges of G_ψ. For $\psi = \prod_{t=1}^{\ell} \delta_{r_t,i_t}\delta_{r_t,j_t}\sigma_{r_t,i_t}\sigma_{r_t,j_t}u_{i_t}^{k_t}u_{j_t}^{k_{t+1}}$ we draw 4ℓ labeled edges in G_ψ with distinct labels in $[4\ell]$. For each $t \in [\ell]$ we draw an edge from the tth left vertex to i_t with label $4(t-1)+1$, from i_t to r_t with label $4(t-1)+2$, from r_t to j_t with label $4(t-1)+3$, and from j_t to the $(t+1)$st left vertex with label $4(t-1)+4$. Many different monomials ψ will map to the same graph G_ψ; in particular the graph maintains no information concerning equalities amongst the k_t, and the y middle vertices may map to any y distinct values in $[n]$, and the right vertices to any z distinct values in $[m]$. We handle the right hand side of Eq. (3) by grouping monomials ψ mapping to the same G, bounding the total contribution of G in terms of its graph structure when summing all ψ with $G_\psi = G$, then summing contributions over all G.

Before continuing further we introduce some more notation then make a few observations. For a graph G as above, recall G has 4ℓ edges, and we refer to the *distinct* edges (ignoring labels) as *bonds*. We let $E(G)$ denote the edge multiset of a multigraph G and $B(G)$ denote the bond set. We refer to the number of bonds a vertex is incident upon as its *bond-degree*, and the number of edges as its *edge-degree*. We do not count self-loops for calculating bond-degree, and we count them twice for edge-degree. We let $LM(G)$ be the induced multigraph on the left and middle vertices of G, and $MR(G)$ be the induced multigraph on the middle and right vertices. We let $w = w(G)$ be the number of connected components in $MR(G)$. We let $b = b(G)$ denote the number of bonds in $MR(G)$ (note $MR(G)$ has 2ℓ edges, but it may happen that $b < 2\ell$ since G is a multigraph). Given G we define the undirected *dot product multigraph* $\widehat{G}$ with vertex set $[y]$. Note every left vertex of G has edge-degree 2. For each $t \in [\ell]$ an edge (i, j) is drawn in $\widehat{G}$ between the two middle vertices that the tth left vertex is adjacent to (we draw a self-loop on i if $i = j$). We label the edges of $\widehat{G}$ according to the natural tour on G (by following edges in increasing label order), and the vertices with distinct labels in $[y]$ in increasing order of when each vertex was first visited by the same tour. We name $\widehat{G}$ the dot product multigraph since if some left vertex t has its two edges connecting to vertices $i, j \in [n]$, then summing over $k_t \in [d]$ produces the dot product $\langle u_i, u_j \rangle$.

Now we make some observations. Due to the random signs $\sigma_{r,i}$, a monomial ψ has expectation zero unless every bond in $MR(G)$ has even multiplicity, in which case the product of random signs is 1. Also, note the expected product of the $\delta_{r,i}$ is at most $(s/m)^b$ by OSNAP properties. Thus letting $\mathcal{G}$ be the set of all such graphs G with even bond multiplicity in $MR(G)$ that arise from some monomial ψ appearing in Eq. (3), and letting $\mathbf{i} = (i_1, j_1, \dots, i_\ell, j_\ell)$, $\mathbf{k} = (k_1, \dots, k_\ell)$, $\mathbf{r} = (r_1, \dots, r_\ell)$,

$$\begin{aligned}
\mathbb{E}\mathrm{tr}((S-I)^\ell) &= \frac{1}{s^\ell} \sum_{G \in \mathcal{G}} \sum_{\substack{\mathbf{i},\mathbf{r} \\ G_{(\mathbf{i},\mathbf{r})} = G}} \left(\mathbb{E} \prod_{t=1}^{\ell} \delta_{r_t,i_t}\delta_{r_t,j_t} \right) \cdot \sum_{\mathbf{k}} \prod_{t=1}^{\ell} u_{i_t}^{k_t} u_{j_t}^{k_{t+1}} \\
&= \frac{1}{s^\ell} \sum_{G \in \mathcal{G}} \sum_{\substack{\mathbf{i},\mathbf{r} \\ G_{(\mathbf{i},\mathbf{r})} = G}} \left(\mathbb{E} \prod_{t=1}^{\ell} \delta_{r_t,i_t}\delta_{r_t,j_t} \right) \cdot \prod_{t=1}^{\ell} \langle u_{i_{t+1}}, u_{j_t} \rangle
\end{aligned}$$

$$= \frac{1}{s^\ell} \sum_{G \in \mathcal{G}} \sum_{\mathbf{i}} \prod_{t=1}^{\ell} \langle u_{i_{t+1}}, u_{j_t} \rangle \cdot \left(\sum_{\substack{\mathbf{r} \\ G_{(\mathbf{i},\mathbf{r})} = G}} \mathbb{E} \prod_{t=1}^{\ell} \delta_{r_t, i_t} \delta_{r_t, j_t} \right)$$

$$\leq \frac{1}{s^\ell} \cdot \sum_{G \in \mathcal{G}} \left(\frac{s}{m} \right)^b \cdot m^z \cdot \left| \sum_{\substack{a_1, \ldots, a_y \in [n] \\ \forall i \neq j \ a_i \neq a_j}} \prod_{\substack{e \in E(\widehat{G}) \\ e = (i,j)}} \langle u_{a_i}, u_{a_j} \rangle \right| \tag{4}$$

where in the above expressions we treat $i_{\ell+1}$ as i_1 and $k_{\ell+1}$ as k_1.

It will also be convenient to introduce a notion we will use in our analysis called a *generalized dot product multigraph*. Such a graph $\widehat{G}$ is just as in the case of a dot product multigraph, except that each edge $e = (i, j)$ is associated with some matrix M_e. We call M_e the *edge-matrix* of e. Furthermore, for an edge $e = (i, j)$ with edge-matrix M_e, we also occasionally view e as the edge (j, i), in which case we say its associated edge-matrix is M_e^*. We then associate with $\widehat{G}$ the product

$$\prod_{\substack{e \in \widehat{G} \\ e = (i,j)}} \langle u_{a_i}, M_e u_{a_j} \rangle.$$

Note that a dot product multigraph is simply a generalized dot product multigraph in which $M_e = I$ for all e. Also, in such a generalized dot product multigraph, we treat multiedges as representing the same bond iff the associated edge-matrices are equal (multiedges may have different edge-matrices).

Lemma 11. *Let H be a connected generalized dot product multigraph on vertex set $[N]$ with $E(H) \neq \emptyset$ and where every bond has even multiplicity. Also suppose that for all $e \in E(H)$, $\|M_e\| \leq 1$. Then*

$$\sum_{a_2=1}^{n} \cdots \sum_{a_N=1}^{n} \prod_{\substack{e \in E(H) \\ e = (i,j)}} \langle v_{a_i}, M_e v_{a_j} \rangle \leq \|c\|^2,$$

where $v_{a_i} = u_{a_i}$ for $i \neq 1$, and v_{a_1} equals some fixed vector c with $\|c\| \leq 1$.

Proof. Let π be some permutation of $\{2, \ldots, N\}$. For a bond $q = (i, j) \in B(H)$, let $2\alpha_q$ denote the multiplicity of q in H. Then by ordering the assignments of the a_t in the summation

$$\sum_{a_2, \ldots, a_N \in [n]} \prod_{\substack{e \in E(H) \\ e = (i,j)}} \langle v_{a_i}, M_e v_{a_j} \rangle$$

according to π, we obtain the exactly equal expression

$$\sum_{a_{\pi(N)}=1}^{n} \prod_{\substack{q \in B(H) \\ q = (\pi(N), j) \\ N \leq \pi^{-1}(j)}} \langle v_{a_{\pi(N)}}, M_q v_{a_j} \rangle^{2\alpha_q} \cdots \sum_{a_{\pi(2)}=1}^{n} \prod_{\substack{q \in B(H) \\ q = (\pi(1), j) \\ 2 \leq \pi^{-1}(j)}} \langle v_{a_{\pi(2)}}, M_q v_{a_j} \rangle^{2\alpha_q}. \tag{5}$$

Here we have taken the product over $t \leq \pi^{-1}(j)$ as opposed to $t < \pi^{-1}(j)$ since there may be self-loops. By Lemma 6 and the fact that $\|c\| \leq 1$ we have that for any i, j, $\langle v_i, v_j \rangle^2 \leq \|v_i\|^2 \cdot \|v_j\|^2 \leq 1$,

so we obtain an upper bound on Eq. (5) by replacing each $\langle v_{a_{\pi(t)}}, v_{a_j}\rangle^{2\alpha_v}$ term with $\langle v_{a_{\pi(t)}}, v_{a_j}\rangle^2$. We can thus obtain the sum

$$\sum_{a_{\pi(N)}=1}^{n} \prod_{\substack{q\in B(H)\\ q=(\pi(N),j)\\ q\le\pi^{-1}(j)}} \langle v_{a_{\pi(N)}}, M_q v_{a_j}\rangle^2 \cdots \sum_{a_{\pi(2)}=1}^{n} \prod_{\substack{q\in B(H)\\ q=(\pi(2),j)\\ 2\le\pi^{-1}(j)}} \langle v_{a_{\pi(2)}}, M_q v_{a_j}\rangle^2, \tag{6}$$

which upper bounds Eq. (5). Now note for $2 \le t \le N$ that for any nonnegative integer β_t and for $\{q \in B(H) : q = (\pi(t), j), t < \pi^{-1}(j)\}$ non-empty (note the strict inequality $t < \pi^{-1}(j)$),

$$\begin{aligned}
\sum_{a_{\pi(t)}=1}^{n} \|v_{a_{\pi(t)}}\|^{2\beta_t} \cdot \prod_{\substack{q\in B(H)\\ q=(\pi(t),j)\\ t\le\pi^{-1}(j)}} \langle v_{a_{\pi(t)}}, M_q v_{a_j}\rangle^2 &\le \sum_{a_{\pi(t)}=1}^{n} \prod_{\substack{q\in B(H)\\ q=(\pi(t),j)\\ t\le\pi^{-1}(j)}} \langle v_{a_{\pi(t)}}, M_q v_{a_j}\rangle^2 \qquad (7)\\
&\le \prod_{\substack{q\in B(H)\\ q=(\pi(t),j)\\ t<\pi^{-1}(j)}} \left(\sum_{a_{\pi(t)}=1}^{n} \langle v_{a_{\pi(t)}}, M_q v_{a_j}\rangle^2 \right)\\
&= \prod_{\substack{q\in B(H)\\ q=(\pi(t),j)\\ t<\pi^{-1}(j)}} \left(\sum_{a_{\pi(t)}=1}^{n} v_{a_j}^* M_q^* v_{a_{\pi(t)}} v_{a_{\pi(t)}}^* M_q v_{a_j} \right)\\
&= \prod_{\substack{q\in B(H)\\ q=(\pi(t),j)\\ t<\pi^{-1}(j)}} (M_q v_{a_j})^* \left(\sum_{i=1}^{n} u_i u_i^* \right) M_q v_{a_j}\\
&= \prod_{\substack{q\in B(H)\\ q=(\pi(t),j)\\ t<\pi^{-1}(j)}} \|M_q v_{a_j}\|^2 \qquad (8)\\
&\le \prod_{\substack{q\in B(H)\\ q=(\pi(t),j)\\ t<\pi^{-1}(j)}} \|v_{a_j}\|^2, \qquad (9)
\end{aligned}$$

where Eq. (7) used Lemma 6, Eq. (8) used Lemma 5, and Eq. (9) used that $\|M_q\| \le 1$. Now consider processing the alternating sum-product in Eq. (6) from right to left. We say that a bond $(i, j) \in B(H)$ is *assigned to* i if $\pi^{-1}(i) < \pi^{-1}(j)$. When arriving at the tth sum-product and using the upper bound Eq. (8) on the previous $t - 1$ sum-products, we will have a sum over $\|v_{a_{\pi(t)}}\|^2$ raised to some nonnegative power (specifically the number of bonds incident upon $\pi(t)$ but not assigned to $\pi(t)$, plus one if $\pi(t)$ has a self-loop) multiplied by a product of $\langle v_{a_{\pi(t)}}, v_{a_j}\rangle^2$ over all bonds $(\pi(t), j)$ assigned to $\pi(t)$. There are two cases. In the first case $\pi(t)$ has no bonds assigned to it. We will ignore this case since we will show that we can choose π to avoid it.

The other case is that $\pi(t)$ has at least one bond assigned to it. In this case we are in the scenario of Eq. (8) and thus summing over $a_{\pi(t)}$ yields a non-empty product of $\|v_{a_j}\|^2$ for the j for

which $(\pi(t), j)$ is a bond assigned to $\pi(t)$. Thus in our final sum, as long as we choose π to avoid the first case, we are left with an upper bound of $\|c\|$ raised to some power equal to the edge-degree of vertex 1 in H, which is at least 2. The lemma would then follow since $\|c\|^j \leq \|c\|^2$ for $j \geq 2$.

It now remains to show that we can choose π to avoid the first case where some $t \in \{2, \ldots, N\}$ is such that $\pi(t)$ has no bonds assigned to it. Let T be a spanning tree in H rooted at vertex 1. We then choose any π with the property that for any $i < j$, $\pi(i)$ is not an ancestor of $\pi(j)$ in T. This can be achieved, for example, by assigning π values in reverse breadth first search order. ■

Lemma 12. *Let $\widehat{G}$ be any dot product graph as in Eq. (4). Then*

$$\left| \sum_{\substack{a_1,\ldots,a_y \in [n] \\ \forall i \neq j \;\; a_i \neq a_j}} \prod_{\substack{e \in \widehat{G} \\ e=(i,j)}} \langle u_{a_i}, u_{a_j} \rangle \right| \leq y! \cdot d^{y-w+1}.$$

Proof. We first note that we have the inequality

$$\left| \sum_{\substack{a_1,\ldots,a_y \in [n] \\ \forall i \neq j \;\; a_i \neq a_j}} \prod_{\substack{e \in E(\widehat{G}) \\ e=(i,j)}} \langle u_{a_i}, u_{a_j} \rangle \right| = \left| \sum_{\substack{a_1,\ldots,a_{y-1} \in [n] \\ \forall i \neq j \in [y-1] \;\; a_i \neq a_j}} \left(\sum_{a_y=1}^{n} \prod_{\substack{e \in E(\widehat{G}) \\ e=(i,j)}} \langle u_{a_i}, u_{a_j} \rangle - \sum_{t=1}^{y-1} \sum_{a_y = a_t} \prod_{\substack{e \in E(\widehat{G}) \\ e=(i,j)}} \langle u_{a_i}, u_{a_j} \rangle \right) \right|$$

$$\leq \left| \sum_{\substack{a_1,\ldots,a_{y-1} \in [n] \\ \forall i \neq j \in [y-1] \;\; a_i \neq a_j}} \sum_{a_y=1}^{n} \prod_{\substack{e \in E(\widehat{G}) \\ e=(i,j)}} \langle u_{a_i}, u_{a_j} \rangle \right| + \sum_{t=1}^{y-1} \left| \sum_{\substack{a_1,\ldots,a_{y-1} \in [n] \\ \forall i \neq j \in [y-1] \;\; a_i \neq a_j}} \sum_{a_y = a_t} \prod_{\substack{e \in E(\widehat{G}) \\ e=(i,j)}} \langle u_{a_i}, u_{a_j} \rangle \right|$$

We can view the sum over t on the right hand side of the above as creating $t-1$ new dot product multigraphs, each with one fewer vertex where we eliminated vertex y and associated it with vertex t for some t, and for each edge (y, a) we effectively replaced it with (t, a). Also in first sum where we sum over all n values of a_y, we have eliminated the constraints $a_y \neq a_i$ for $i \neq y$. By recursively applying this inequality to each of the resulting t summations, we bound

$$\left| \sum_{\substack{a_1,\ldots,a_y \in [n] \\ \forall i \neq j \;\; a_i \neq a_j}} \prod_{\substack{e \in E(\widehat{G}) \\ e=(i,j)}} \langle u_{a_i}, u_{a_j} \rangle \right|$$

by a sum of contributions from $y!$ dot product multigraphs where in none of these multigraphs do we have the constraint that $a_i \neq a_j$ for $i \neq j$. We will show that each one of these resulting multigraphs contributes at most d^{y-w+1}, from which the lemma follows.

Let G' be one of the dot product multigraphs at a leaf of the above recursion so that we now wish to bound

$$F(G') \stackrel{\text{def}}{=} \left| \sum_{a_1,\ldots,a_y=1}^{n} \prod_{\substack{e \in E(G') \\ e=(i,j)}} \langle u_{a_i}, M_e u_{a_j} \rangle \right| \tag{10}$$

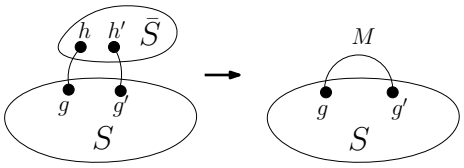

Figure 2: The formation of H_t from H_{t-1}.

where $M_e = I$ for all e for G'. Before proceeding, we first claim that every connected component of G' is Eulerian. To see this, observe G has an Eulerian tour, by following the edges of G in increasing order of label, and thus all middle vertices have even edge-degree in G. However they also have even edge-degree in $MR(G)$, and thus the edge-degree of a middle vertex in $LM(G)$ must be even as well. Thus, every vertex in $\widehat{G}$ has even edge-degree, and thus every vertex in each of the recursively created leaf graphs also has even edge-degree since at every step when we eliminate a vertex, some other vertex's degree increases by the eliminated vertex's degree which was even. Thus every connected component of G' is Eulerian as desired.

We now upper bound $F(G')$. Let the connected components of G' be $C_1, \ldots, C_{CC(G')}$, where $CC(\cdot)$ counts connected components. An observation we repeatedly use later is that for any generalized dot product multigraph H with components $C_1, \ldots, C_{CC(H)}$,

$$F(H) = \prod_{i=1}^{CC(H)} F(C_i). \tag{11}$$

We treat G' as a generalized dot product multigraph so that each edge e has an associated matrix M_e (though in fact $M_e = I$ for all e). Define an undirected multigraph to be *good* if all its connected components have two edge-disjoint spanning trees. We will show that $F(G') \leq F(G'')$ for some generalized dot product multigraph G'' that is good then will show $F(G'') \leq d^{y-w+1}$. If G' itself is good then we can set $G'' = G'$. Otherwise, we will show $F(G') = F(H_0) = \ldots = F(H_\tau)$ for smaller and smaller generalized dot product multigraphs H_t (i.e. with successively fewer vertices) whilst maintaining the invariant that each H_t has Eulerian connected components and has $\|M_e\| \leq 1$ for all e. We stop when some H_τ is good and we can set $G'' = H_\tau$.

Let us now focus on constructing this sequence of H_t in the case that $\widehat{G}$ is not good. Let $H_0 = \widehat{G}$. Suppose we have constructed $H_0, \ldots, H_{t-1}$ for $i \geq 1$ none of which are good, and now we want to construct H_t. Since H_{t-1} is not good it cannot be 4-edge-connected by Corollary 8, so there is some connected component C_{j^*} of H_{t-1} with some cut $S \subsetneq V(C_{j^*})$ with 2 edges crossing the cut $(S, \bar{S})$, where $\bar{S}$ represents the complement of S in C_{j^*}. This is because since C_{j^*} is Eulerian, any cut has an even number of edges crossing it. Choose such an $S \subset V(C_{j^*})$ with $|\bar{S}|$ minimum amongst all such cuts. Let the two edges crossing the cut be $(g,h), (g',h')$ with $g, g' \in S$ (note that it may be the case that $g = g'$ or $h = h'$, or both). Note that $F(H_{t-1})$ equals the magnitude of

$$\sum_{\substack{\{a_i\} \\ i \notin C_{j^*}}} \prod_{\substack{e \in H_{t-1} \\ e \notin C_{j^*} \\ e=(i,j)}} \langle u_{a_i}, M_e u_{a_j} \rangle \sum_{\substack{\{a_i\} \\ i \in S}} \left(\prod_{\substack{e \in H_{t-1} \\ e=(i,j) \\ i,j \in S}} \langle u_{a_i}, M_e u_{a_j} \rangle \right) u^*_{a_g} \underbrace{\left(\sum_{\substack{\{a_i\} \\ i \in \bar{S}}} M_{(g,h)} u_{a_h} \left(\prod_{\substack{e \in H_{t-1} \\ e=(i,j) \\ i,j \in \bar{S}}} \langle u_{a_i}, M_e u_{a_j} \rangle \right) u^*_{a_{h'}} M_{(h',g')} \right)}_{M} u_{a_{g'}}. \tag{12}$$

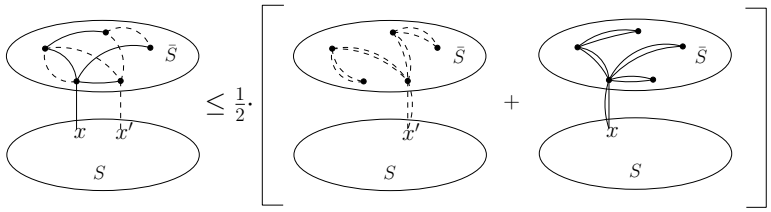

Figure 3: Showing that $\|M\| \leq 1$ by AM-GM on two edge-disjoint spanning subgraphs.

In the above summations over $\{a_i\}$ we also have the constraints that $a_i \neq a_j$ for $i \neq j$. We define H_t to be H_{t-1} but where in the j^*th component we eliminate all the vertices and edges in $\bar{S}$ and add an additional edge from g to g' which we assign edge-matrix M (see Figure 2). We thus have that $F(H_{t-1}) = F(H_t)$. Furthermore each component of H_t is still Eulerian since every vertex in H_{t-1} has either been eliminated, or its edge-degree has been preserved and thus all edge-degrees are even. By iteratively eliminating bad cuts $\bar{S}$ in this way, we eventually arrive at a generalized dot product multigraph H_τ that has two edge-disjoint spanning trees in every component; this is because this iterative process terminates, since every successive H_t has at least one fewer vertex, and when the number of vertices of any connected component drops to 2 or lower then that connected component has two edge-disjoint spanning trees.

We first claim that $C_{j^*}(\bar{S})$ has two edge-disjoint spanning trees. Define C' to be the graph $C_{j^*}(\bar{S})$ with an edge from h to h' added. We show that $C'(\bar{S})$ is 4-edge-connected so that $C_{j^*}(\bar{S})$ has two edge-disjoint spanning trees by Corollary 8. Now to see this, consider some $S' \subsetneq \bar{S}$. Consider the cut $(S', V(C')\backslash S')$. C' is Eulerian, so the number of edges crossing this cut is either 2 or at least 4. If it 2, then since $|S'| < |\bar{S}|$ this is a contradiction since $\bar{S}$ was chosen amongst such cuts to have $|\bar{S}|$ minimum. Thus it is at least 4, and we claim that the number of edges crossing the cut $(S', \bar{S}\backslash S')$ in $C'(\bar{S})$ must also be at least 4. If not, then it is 2 since $C'(\bar{S})$ is Eulerian. However since the number of edges leaving S' in C' is at least 4, it must then be that $h, h' \in S'$. But then the cut $(\bar{S}\backslash S', V(C')\backslash(\bar{S}\backslash S'))$ has 2 edges crossing it so that $\bar{S}\backslash S'$ is a smaller cut than $\bar{S}$ with 2 edges leaving it in C', violating the minimality of $|\bar{S}|$, a contradiction. Thus $C'(\bar{S})$ is 4-edge-connected, implying $C_{j^*}(\bar{S})$ has two edge-disjoint spanning trees T_1, T_2 as desired.

Now to show $\|M\| \leq 1$, by Fact 9 we have $\|M\| = \sup_{\|x\|,\|x'\|=1} x^* M x'$. We have (see Figure 3)

$$
\begin{aligned}
x^* M x' &= \sum_{a_S \in [n]^{|S|}} \langle x, M_{(g,h)} u_{a_h} \rangle \cdot \left(\prod_{\substack{e \in E(C_{j^*}(S)) \\ e=(i,j)}} \langle u_{a_i}, M_e u_{a_j} \rangle \right) \cdot \langle u_{a_{h'}}, M_{(h',g')} x' \rangle \\
&= \sum_{a_S \in [n]^{|S|}} \left(\langle x, M_{(g,h)} u_{a_h} \rangle \cdot \prod_{\substack{e \in T_1 \\ e=(i,j)}} \langle u_{a_i}, M_e u_{a_j} \rangle \right) \cdot \left(\langle u_{a_{h'}}, M_{(h',g')} x' \rangle \cdot \prod_{\substack{e \in E(C_{j^*}(S))\backslash T_1 \\ e=(i,j)}} \langle u_{a_i}, M_e u_{a_j} \rangle \right) \\
&\leq \frac{1}{2} \cdot \left[\sum_{a_S \in [n]^{|S|}} \left(\langle x, M_{(g,h)} u_{a_h} \rangle^2 \cdot \prod_{\substack{e \in T_1 \\ e=(i,j)}} \langle u_{a_i}, M_e u_{a_j} \rangle^2 \right) \right.
\end{aligned}
$$

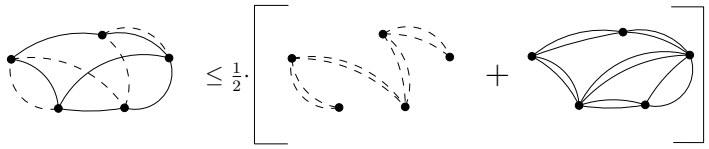

Figure 4: AM-GM on two edge-disjoint spanning subgraphs of one connected component of G''.

$$+ \sum_{a_S \in [n]^{|S|}} \left(\langle u_{a_{h'}}, M_{(h',g')} x' \rangle^2 \cdot \prod_{\substack{e \in E(C_{j^*}(S)) \setminus T_1 \\ e=(i,j)}} \langle u_{a_i}, M_e u_{a_j} \rangle^2 \right) \Bigg] \tag{13}$$

$$\leq \frac{1}{2} \left(\|x\|^2 + \|x'\|^2 \right) \tag{14}$$

$$= 1,$$

where Eq. (13) used the AM-GM inequality, and Eq. (14) used Lemma 11 (note the graph with vertex set $S \cup \{g'\}$ and edge set $E(C_{j^*}(S)) \setminus T_1 \cup \{(g', h')\}$ is connected since $T_2 \subseteq E(C_{j^*}(S)) \setminus T_1$). Thus we have shown that H_t satisfies the desired properties. Now notice that the sequence $H_0, \ldots, H_1, \ldots$ must eventually terminate since the number of vertices is strictly decreasing in this sequence and any Eulerian graph on 2 vertices is good. Therefore we have that H_τ is eventually good for some $\tau > 0$ and we can set $G'' = H_\tau$.

It remains to show that for our final good G'' we have $F(G'') \leq d^{y-w+1}$. We will show this in two parts by showing that both $CC(G'') \leq d^{y-w+1}$ and $F(G'') \leq d^{CC(G'')}$. For the first claim, note that $CC(G'') \leq CC(\widehat{G})$ since every H_t has the same number of connected components as G', and $CC(G') \leq CC(\widehat{G})$. This latter inequality holds since in each level of recursion used to eventually obtain G' from $\widehat{G}$, we repeatedly identified two vertices as equal and merged them, which can only decrease the number of connected components. Now, all middle vertices in G lie in one connected component (since G is connected) and $MR(G)$ has w connected components. Thus the at least $w-1$ edges connecting these components in G must come from $LM(G)$, implying that $LM(G)$ (and thus $\widehat{G}$) has at most $y-w+1$ connected components, which thus must also be true for G'' as argued above.

It only remains to show $F(G'') \leq d^{CC(G'')}$. Let G'' have connected components $C_1, \ldots, C_{CC(G'')}$ with each C_j having 2 edge-disjoint spanning trees T_1^j, T_2^j (see Figure 4). We then have

$$\begin{aligned} F(G'') &= \prod_{t=1}^{CC(G'')} F(C_t) \\ &= \prod_{t=1}^{CC(G'')} \left| \sum_{a_1,\ldots,a_{|V(C_t)|}=1}^{n} \prod_{\substack{e \in E(C_t) \\ e=(i,j)}} \langle u_{a_i}, M_e u_{a_j} \rangle \right| \end{aligned}$$

$$
\begin{aligned}
&= \prod_{t=1}^{CC(G'')} \left| \sum_{a_1,\ldots,a_{|V(C_t)|}=1}^{n} \left(\prod_{\substack{e\in T_1^t \\ e=(i,j)}} \langle u_{a_i}, M_e u_{a_j}\rangle \right) \cdot \left(\prod_{\substack{e\in E(C_t)\setminus T_1^t \\ e=(i,j)}} \langle u_{a_i}, M_e u_{a_j}\rangle \right) \right| \\
&\le \prod_{t=1}^{CC(G'')} \frac{1}{2} \left[\sum_{a_1=1}^{n} \sum_{a_2,\ldots,a_{|V(C_t)|}=1}^{n} \prod_{\substack{e\in T_1^t \\ e=(i,j)}} \langle u_{a_i}, M_e u_{a_j}\rangle^2 + \sum_{a_1=1}^{n} \sum_{a_2,\ldots,a_{|V(C_t)|}=1}^{n} \prod_{\substack{e\in E(C_t)\setminus T_1^t \\ e=(i,j)}} \langle u_{a_i}, M_e u_{a_j}\rangle^2 \right] \qquad (15) \\
&\le \prod_{t=1}^{CC(G'')} \sum_{a_1=1}^{n} \|u_{a_1}\|^2 \qquad (16) \\
&= \prod_{t=1}^{CC(G'')} \|U\|_F^2 \\
&= d^{CC(G'')}
\end{aligned}
$$

where Eq. (15) used the AM-GM inequality, and Eq. (16) used Lemma 11, which applies since $V(C_t)$ with edge set T_1^t is connected, and $V(C_t)$ with edge set $E(C_t)\setminus T_1^t$ is connected (since $T_2^t \subseteq E(C_t)\setminus T_1^t$). ■

Now, for any $G \in \mathcal{G}$ we have $y + z \le b + w$ since for any graph the number of edges plus the number of connected components is at least the number of vertices. We also have $b \ge 2z$ since every right vertex of G is incident upon at least two distinct bonds (since $i_t \ne j_t$ for all t). We also have $y \le b \le \ell$ since $MR(G)$ has exactly 2ℓ edges with no isolated vertices, and every bond has even multiplicity. Finally, a crude bound on the number of different $G \in \mathcal{G}$ with a given b, y, z is $(zy^2)^\ell / y!z! \le (b^3)^\ell / y!$. This is because of the following reason. Label the y middle vertices $1, \ldots, y$ and the z right vertices $1, \ldots, z$. Let the vertices be numbered in increasing order, ordered by the first time visited. When drawing the graph edges in increasing order of edge label, when at a left vertex, we draw edges from the left to the middle, then to the right, then to the middle, and then back to the left again, giving $y^2 z$ choices. This is done ℓ times. We can divide by $y!, z!$ since counting graphs in this way overcounts each graph $y!z!$ times, since the order in which we visit vertices might not be consistent with their labelings. Thus by Lemma 12 and Eq. (4),

$$
\begin{aligned}
\mathbb{E}\mathrm{tr}((S-I)^\ell) &\le d \cdot \frac{1}{s^\ell} \sum_{b,y,z,w} \sum_{\substack{G\in\mathcal{G} \\ b(G)=b, y(G)=y \\ w(G)=w, z(G)=z}} y! \cdot s^b \cdot m^{z-b} \cdot d^{y-w} \\
&\le d \cdot \frac{1}{s^\ell} \sum_{b,y,z,w} y! \cdot s^b \sum_{\substack{G\in\mathcal{G} \\ b(G)=b, y(G)=y \\ w(G)=w, z(G)=z}} \left(\frac{d}{m}\right)^{b-z} \\
&\le d \cdot \frac{1}{s^\ell} \sum_{b,y,z,w} b^{3\ell} s^b \cdot \left(\frac{d}{m}\right)^{b-z}
\end{aligned}
$$

$$\leq d \cdot \frac{1}{s^\ell} \sum_{b,y,z,w} b^{3\ell} \left(s\sqrt{\frac{d}{m}}\right)^b$$

$$\leq d\ell^4 \cdot \max_{2\leq b\leq \ell} \left(\frac{b^3}{s}\right)^{\ell-b} \left(b^3\sqrt{\frac{d}{m}}\right)^b \quad (17)$$

Define $\epsilon = 2\varepsilon - \varepsilon^2$. For $\ell \geq \ln(d\ell^4/\delta) = O(\ln(d/\delta))$, $s \geq e\ell^3/\epsilon = O(\log(d/\delta)^3/\varepsilon)$, and $m \geq e^2 d\ell^6/\epsilon^2 = O(d\log(d/\delta)^6/\varepsilon^2)$, the above expression is at most $\delta\epsilon^\ell$. Thus by Eq. (1),

$$\mathbb{P}\left(\|S - I\| > \epsilon\right) < \frac{1}{\epsilon^\ell} \cdot \mathbb{E}\mathrm{tr}((S-I)^\ell) \leq \delta.$$

■

The proof of Theorem 10 reveals that for $\delta = 1/\mathrm{poly}(d)$ one could also set $m = O(d^{1+\gamma}/\varepsilon^2)$ and $s = O_\gamma(1/\varepsilon)$ for any fixed constant $\gamma > 0$ and arrive at the same conclusion. Indeed, let $\gamma' < \gamma$ be any positive constant. Let ℓ in the proof of Theorem 10 be taken as $O(\log(d/\delta)) = O(\log d)$. It suffices to ensure $\max_{2\leq b\leq \ell}(b^3/s)^{\ell-b} \cdot (b^3\sqrt{d/m})^b \leq \varepsilon^\ell\delta/(ed\ell^4)$ by Eq. (17). Note $d^{\gamma' b/2} > b^{3\ell}$ as long as $b/\ln b > 6\gamma^{-1}\ell/\ln d = O(1/\gamma')$, so $d^{\gamma' b} > b^{3\ell}$ for $b > b^*$ for some $b^* = \Theta(\gamma^{-1}\log(1/\gamma))$. We choose $s \geq e(b^*)^3/\varepsilon$ and $m = d^{1+\gamma}/\varepsilon^2$, which is at least $d^{1+\gamma'}\ell^6/\varepsilon^2$for d larger than some fixed constant. Thus the max above is always as small as desired, which can be seen by looking at $b \leq b^*$ and $b > b^*$ separately (in the former case $b^3/s < 1/e$, and in the latter case $(b^3/s)^{\ell-b} \cdot (b^3\sqrt{d/m})^b < (\varepsilon/e)^\ell b^{3\ell} d^{-\gamma' b/2} = (\varepsilon/e)^\ell e^{3\ell\ln b - (1/2)\gamma' b\ln d} < (\varepsilon/e)^\ell$ is as small as desired). This observation yields:

Theorem 13. *Let $\alpha, \gamma > 0$ be arbitrary constants. For Π an OSNAP with $s = \Theta(1/\varepsilon)$ and $\varepsilon \in (0,1)$, with probability at least $1 - 1/d^\alpha$, all singular values of ΠU are $1 \pm \varepsilon$ for $m = \Omega(d^{1+\gamma}/\varepsilon^2)$ and σ, h being $\Omega(\log d)$-wise independent. The constants in the big-Θ and big-Ω depend on α, γ.*

Remark 14. Section 1 stated the time to list all non-zeroes in a column in Theorem 10 is $t_c = \tilde{O}(s)$. For $\delta = 1/\mathrm{poly}(d)$, naively one would actually achieve $t_c = O(s \cdot \log d)$ since one needs to evaluate an $O(\log d)$-wise independent hash function s times. This can be improved to $\tilde{O}(s)$ using fast multipoint evaluation of hash functions; see for example the last paragraph of Remark 16 of [27].

3 Applications

We use the fact that many matrix problems have the same time complexity as matrix multiplication including computing the matrix inverse [8] [22, Appendix A], and QR decomposition [41]. In this paper we only consider the real RAM model and state the running time in terms of the number of field operations. The algorithms for solving linear systems, computing inverse, QR decomposition, and approximating SVD based on fast matrix multiplication can be implemented with precision comparable to that of conventional algorithms to achieve the same error bound (with a suitable notion of approximation/stability). We refer readers to [16] for details. Notice that it is possible that both algorithms based on fast matrix multiplication and conventional counterparts are unstable, see e.g. [5] for an example of a pathological matrix with very high condition number.

In this section we describe some applications of our subspace embeddings to problems in numerical linear algebra. All applications follow from a straightforward replacement of previously used embeddings with our new ones as most proofs go through verbatim. In the statement of our bounds we implicitly assume $\mathrm{nnz}(A) \geq n$, since otherwise fully zero rows of A can be ignored without affecting the problem solution.

3.1 Approximate Leverage Scores

This section describes the application of our subspace embedding from Theorem 10 or Theorem 13 to approximating the leverage scores. Consider a matrix A of size $n \times d$ and rank r. Let U be a $n \times r$ matrix whose columns form an orthonormal basis of the column space of A. The *leverage scores* of A are the squared lengths of the rows of U. The algorithm for approximating the leverage scores and the analysis are the same as those of [13], which itself uses essentially the same algorithm outline as Algorithm 1 of [17]. The improved bound is stated below (cf. [13, Theorem 29]).

Theorem 15. *For any constant $\varepsilon > 0$, there is an algorithm that with probability at least $2/3$, approximates all leverage scores of a $n \times d$ matrix A in time $\tilde{O}(\mathrm{nnz}(A)/\varepsilon^2 + r^\omega \varepsilon^{-2\omega})$.*

Proof. As in [13], this follows by replacing the Fast Johnson-Lindenstrauss embedding used in [17] with our sparse subspace embeddings. The only difference is in the parameters of our OSNAPs. We essentially repeat the argument verbatim just to illustrate where our new OSE parameters fit in; nothing in this proof is new. Now, we first use [10] so that we can assume A has only $r = \mathrm{rank}(A)$ columns and is of full column rank. Then, we take an OSNAP Π with $m = \tilde{O}(r/\varepsilon^2), s = (\mathrm{polylog}\, r)/\varepsilon$ and compute ΠA. We then find R^{-1} so that $\Pi A R^{-1}$ has orthonormal columns. The analysis of [17] shows that the ℓ_2^2 of the rows of AR^{-1} are $1 \pm \varepsilon$ times the leverage scores of A. Take $\Pi' \in \mathbb{R}^{r \times t}$ to be a JL matrix that preserves the ℓ_2 norms of the n rows of AR^{-1} up to $1 \pm \varepsilon$. Finally, compute $R^{-1}\Pi'$ then $A(R^{-1}\Pi')$ and output the squared row norms of $AR\Pi'$.

Now we bound the running time. The time to reduce A to having r linearly independent columns is $O((\mathrm{nnz}(A) + r^\omega) \log n)$. ΠA can be computed in time $O(\mathrm{nnz}(A) \cdot (\mathrm{polylog}\, r)/\varepsilon)$. Computing $R \in \mathbb{R}^{r \times r}$ from the QR decomposition takes time $\tilde{O}(m^\omega) = \tilde{O}(r^\omega/\varepsilon^{2\omega})$, and then R can be inverted in time $\tilde{O}(r^\omega)$; note $\Pi A R^{-1}$ has orthonormal columns. Computing $R^{-1}\Pi'$ column by column takes time $O(r^2 \log r)$ using the FJLT of [4, 32] with $t = O(\varepsilon^{-2} \log n (\log\log n)^4)$. We then multiply the matrix A by the $r \times t$ matrix $R^{-1}\Pi'$, which takes time $O(t \cdot \mathrm{nnz}(A)) = \tilde{O}(\mathrm{nnz}(A)/\varepsilon^2)$. ■

3.2 Least Squares Regression

In this section, we describe the application of our subspace embeddings to the problem of least squares regression. Here given a matrix A of size $n \times d$ and a vector $b \in \mathbb{R}^n$, the objective is to find $x \in \mathbb{R}^d$ minimizing $\|Ax - b\|_2$. The reduction to subspace embedding is similar to those of [13, 40]. The proof is included for completeness.

Theorem 16. *There is an algorithm for least squares regression running in time $O(\mathrm{nnz}(A) + d^3 \log(d/\varepsilon)/\varepsilon^2)$ and succeeding with probability at least $2/3$.*

Proof. Applying Theorem 4 to the subspace spanned by columns of A and b, we get a distribution over matrices Π of size $O(d^2/\varepsilon^2) \times n$ such that Π preserves lengths of vectors in the subspace up to a factor $1 \pm \varepsilon$ with probability at least $5/6$. Thus, we only need to find $\mathrm{argmin}_x \|\Pi A x - \Pi b\|_2$. Note that ΠA has size $O(d^2/\varepsilon^2) \times d$. By Theorem 12 of [40], there is an algorithm that with probability at least $5/6$, finds a $1 \pm \varepsilon$ approximate solution for least squares regression for the smaller input of ΠA and Πb and runs in time $O(d^3 \log(d/\varepsilon)/\varepsilon^2)$. ■

The following theorem follows from using the embedding of Theorem 10 and the same argument as [13, Theorem 40].

Theorem 17. *Let r be the rank of A. There is an algorithm for least squares regression running in time $O(\mathrm{nnz}(A)((\log r)^{O(1)} + \log(n/\varepsilon)) + r^{\omega}(\log r)^{O(1)} + r^2\log(1/\varepsilon))$ and succeeding with probability at least $2/3$.*

3.3 ℓ_p Regression

Given a matrix A of size $n \times d$ and a vector $b \in \mathbb{R}^n$, the ℓ_p regression objective is to find $x \in \mathbb{R}^d$ minimizing $\|Ax-b\|_p$, for some given $p \in [1,\infty)$. A black-box reduction from ℓ_p regression to OSE's was given by [11] using work of [14], and was later pointed out again in [13]. We now describe what our work yields when combined with this reduction.

We first give the following definition from [14].

Definition 18. Let $A \in \mathbb{R}^{n\times d}$ have rank r, and for $p \in [1,\infty)$ let q be such that $1/q + 1/p = 1$. Then $U \in \mathbb{R}^{n\times r}$ is an (α,β,p)-*well-conditioned basis* for A if (1) the columns of U and that of A span the same space, (2) $|||U|||_p \stackrel{\text{def}}{=} \left(\sum_{i,j}|U_{i,j}|^p\right)^{1/p}$ satisfies $|||U|||_p \le \alpha$, and (3) for all $z \in \mathbb{R}^r$ we have $\|z\|_q \le \|Uz\|_p$. We say U is a *p-well-conditioned basis* if α,β are bounded by a polynomial in r, independent of n,d.

Using [10] we can preprocess A in time $\tilde{O}(\mathrm{nnz}(A) + r^{\omega})$ time to remove dependent columns, so we assume that A has full column rank in what follows, i.e. $r = d$. In order to compute an optimal solution up to $1+\varepsilon$, the work of [14] gave a sampling algorithm that, given an (α,β,p)-well-conditioned basis U, produces two new ℓ_p regression problems obtained by sampling rows of A. Solving the first regression problem leads to an 8-approximation, which is refined by solving a second regression problem that leads to a $1+\varepsilon$ error guarantee. Specifically, one first picks sampling probabilities for $i \in [n]$ with $p_i \ge \min\{1, (\|U_i\|_p^p/|||U|||_p^p)\cdot n_1\}$ where U_i is the ith row of U. Then one creates an $n \times n$ diagonal matrix D and sets $D_{i,i}$ to be $1/p_i^{1/p}$ with probability p_i and 0 otherwise then solves the new ℓ_p regression problem of computing $\hat{x} = \mathrm{argmin}_x \|DAx - Db\|_p$. Here n_1 is chosen to be $O(2^p d(\alpha\beta)^p)$. Note that the new ℓ_p regression problem has expected size $n_1 \times d$ as opposed to $n\times d$, and thus can be solved more quickly if α,β are small. The vector $\rho = A\hat{x}-b$ is then used to define new sampling probabilities $q_i = \min\{1, \max\{p_i, (|\rho_i|^p/\|\rho\|_p^p)\cdot n_2\}\}$, which similarly gives a new ℓ_p regression problem with an expected n_2 rows for $n_2 = O(\varepsilon^{-2}24^p d(\alpha\beta)^p\log(1/\varepsilon))$. Let the optimal solution of this second problem be $\hat{x}'$. [14, Theorem 7] showed that $\|A\hat{x}' - b\|_p \le (1+\varepsilon)\cdot\min_x\|Ax-b\|_p$ with probability $2/3$ over all samplings.

The work [11] showed how to use OSE's to speed up the computation of a p-well-conditioned basis, to then implement the above scheme quickly. It follows from [11] (see also [13]) that if one has an OSE distribution with success probability $1-\delta$ for $\delta = 1/n$ (as opposed to $\delta = 1/3$ as in Definition 1), with $\varepsilon = 1/2$, and m rows and column sparsity s to preserve subspaces of dimension d in $R^{n'}$ for $n' = \max\{1, n/d^3\}$, then one can find a matrix U such that AU is a $(\hat{\alpha},\hat{\beta}_m,p)$-well-conditioned basis for A in time $O(\mathrm{nnz}(A)\cdot(s+\log n) + d^3\log n)$. Here $\hat{\alpha} = d^{1/p+1/2}$, $\hat{\beta}_m = O(\max\{1, d^{1/q-1/2}\}\cdot d(m^2d^3)^{|1/p-1/2|})$. Furthermore it is discussed how one can use the Johnson-Lindenstrauss lemma [24] to obtain an approximation to all ℓ_p norms of rows of AU up to a factor of $d^{|1/2-1/p|}$ with probability $1-1/\mathrm{poly}(n)$ in time $O((\mathrm{nnz}(A)+d^2)\log n)$. This approximate knowledge of the row ℓ_p norms leads to a factor $d^{p|1/2-1/p|}$ increase in n_1, n_2 above. One then obtains the following theorem by combining everything stated thus far. This combined statement was also noted in [13], but without explicitly stated dependence on m, s and other parameters; we make this dependence explicit so that we can compare the consequences of using different OSE's.

Theorem 19 (follows from [11,14])**.** *Suppose $A \in \mathbb{R}^{n\times d}$ has rank d. Given an OSE distribution over $\mathbb{R}^{m\times n}$ with column sparsity s, with $\varepsilon = 1/2$ and failure probability $\delta < 1/n$, one can find $\hat{x}' \in \mathbb{R}^d$ in time $O(\mathrm{nnz}(A)(s+\log n)+d^3\log n+\phi(O(2^p d^{1+|p/2-1|}(\hat{\alpha}\hat{\beta}_m)^p, d))+\phi(O(\varepsilon^{-2}24^p d(\hat{\alpha}\hat{\beta}_m)^p \log(1/\varepsilon)), d)$ satisfying $\|A\hat{x}' - b\|_p \le (1+\varepsilon)\min_x \|Ax-b\|_p$ with probability $1/2$. Here $\phi(n,d)$ is the time to exactly solve an $n \times d$ ℓ_p regression problem, and $\hat{\alpha}, \hat{\beta}_m$ are as above.*

The work [13] plugged their OSE with $m = O(d^2 \log n + d\log^2 n) = d^2 \operatorname{polylog} n$ and $s = \log n$ into Theorem 19 above (recall $\hat{\beta}_m$ depends on m^2). On the other hand, one obtains improved dependence on d by using our Theorem 10 with $m = d\operatorname{polylog} n, s = \operatorname{polylog} n$. If n, d are polynomially related one can also use Theorem 13 with $m = O(d^{1+\gamma}), s = O_\gamma(1)$ for any $\gamma > 0$.

3.4 Low Rank Approximation

In this section, we describe the application of our subspace embeddings to low rank approximation. Here given a matrix A, one wants to find a rank k matrix A_k minimizing $\|A - A_k\|_F$. Let Δ_k be the minimum $\|A - A_k\|_F$ over all rank k matrices A_k. We say a distribution $\mathcal{D}$ over $\mathbb{R}^{m\times n}$ has the $(\varepsilon, \delta, \ell)$-moment property if for any $x \in \mathbb{R}^n$ of unit ℓ_2 norm,

$$\mathbb{E}_{\Pi\sim\mathcal{D}} \left|\|\Pi x\|^2 - 1\right|^\ell \le \varepsilon^\ell \cdot \delta.$$

The following was stated in [26, Theorem 5.1] only for the case $\ell = \log(1/\delta)$, but the proof given there works essentially verbatim to provide the following statement.

Theorem 20. *Fix $\varepsilon, \delta > 0$. Suppose a distribution $\mathcal{D}$ over $\mathbb{R}^{m\times n}$ satisfies the $(\varepsilon, \delta, \ell)$-moment property for some $\ell \ge 2$. Then for any matrices A, B with n rows,*

$$\mathbb{P}_{\Pi\sim\mathcal{D}} \left(\|A^T\Pi^T\Pi B - A^T B\|_F > 3\varepsilon/2\|A\|_F\|B\|_F\right) < \delta \tag{18}$$

Any OSNAP with $m = \Omega(1/(\varepsilon^2\delta)), s \ge 1$ satisfies the $(\varepsilon, \delta, 2)$-moment property by the analysis in [43], and thus Theorem 20 is applicable. The reduction from rank-k approximation to OSE's in [13] required one additional property: the subspace embedding matrix also approximates matrix multiplication in the sense of Theorem 20 with error $O(\sqrt{\varepsilon/k})$, which is satisfied by OSNAP with $m = \Omega(k/(\varepsilon\delta))$.

Therefore, the same algorithm and analysis as in [13] work. We state the improved bounds using the embedding of Theorem 4 and Theorem 13 below (cf. [13, Theorem 44]).

Theorem 21. *Given a matrix A of size $n \times n$, there are 2 algorithms that, with probability at least $3/5$, find 3 matrices U, Σ, V where U is of size $n \times k$, Σ is of size $k \times k$, V is of size $n \times k$, $U^TU = V^TV = I_k$, Σ is a diagonal matrix, and*

$$\|A - U\Sigma V^*\|_F \le (1+\varepsilon)\Delta_k$$

The first algorithm runs in time $O(\mathrm{nnz}(A))+\tilde{O}(nk^2+nk^{\omega-1}\varepsilon^{-1-\omega}+k^\omega\varepsilon^{-2-\omega})$. The second algorithm runs in time $O_\gamma(\mathrm{nnz}(A)) + \tilde{O}(nk^{\omega+\gamma-1}\varepsilon^{-1-\omega-\gamma} + k^{\omega+\gamma}\varepsilon^{-2-\omega-\gamma})$ for any constant $\gamma > 0$.

Proof. The proof is essentially the same as that of [13] so we only mention the difference. We use 2 bounds for the running time: multiplying an $a \times b$ matrix and a $b \times c$ matrix with $c > a$ takes $O(a^{\omega-2}bc)$ time (simply dividing the matrices into $a \times a$ blocks), and approximating SVD for an $a \times b$ matrix M with $a > b$ takes $O(ab^{\omega-1})$ time (time to compute M^TM, approximate SVD of

$M^TM = QDQ^T$ in $O(b^\omega)$ time [16], and compute MQ to complete the SVD of M). The running time of [13] comes mainly from the following steps: (1) applying the subspace embedding for rank k/ε to A, (2) applying a sampled Hadamard matrix on a $m \times n$ matrix (m is the number of rows of the subspace embedding matrix), (3) computing the SVD of a $\tilde{O}(k/\varepsilon^3) \times \tilde{O}(k/\varepsilon)$ matrix, (4) multiplying 2 matrices of sizes $\tilde{O}(k/\varepsilon) \times \tilde{O}(k/\varepsilon^3)$ and $\tilde{O}(k/\varepsilon^3) \times n$, and (5) computing the SVD of a $\tilde{O}(k/\varepsilon) \times n$ matrix, hence the terms in the stated running time. The only difference between the two algorithms is that in the first algorithm, the subspace embedding has $m = O(k^2)$ and column sparsity $s = 1$, while in the second algorithm, $m = k^{1+O(\gamma)}$ and $s = O_\gamma(1)$. ■

Acknowledgments

We thank Andrew Drucker for suggesting the SNAP acronym for the OSE's considered in this work, to which we added the "oblivious" descriptor.

References

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

[2] Nir Ailon and Bernard Chazelle. The Fast Johnson–Lindenstrauss transform and approximate nearest neighbors. *SIAM J. Comput.*, 39(1):302–322, 2009.

[3] Nir Ailon and Edo Liberty. Fast dimension reduction using Rademacher series on dual BCH codes. *Discrete Comput. Geom.*, 42(4):615–630, 2009.

[4] Nir Ailon and Edo Liberty. Almost optimal unrestricted fast Johnson-Lindenstrauss transform. In *Proceedings of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 185–191, 2011.

[5] Noga Alon and Van H. Vu. Anti-Hadamard matrices, coin weighing, threshold gates, and indecomposable hypergraphs. *J. Comb. Theory, Ser. A*, 79(1):133–160, 1997.

[6] Z.D. Bai and Y.Q. Yin. Limit of the smallest eigenvalue of a large dimensional sample covariance matrix. *Ann. Probab.*, 21(3):1275–1294, 1993.

[7] Vladimir Braverman, Rafail Ostrovsky, and Yuval Rabani. Rademacher chaos, random Eulerian graphs and the sparse Johnson-Lindenstrauss transform. *CoRR*, abs/1011.2590, 2010.

[8] James R. Bunch and John E. Hopcroft. Triangular factorization and inversion by fast matrix multiplication. *Math. Comp.*, 28:231–236, 1974.

[9] Jian-Feng Cai, Emmanuel J. Candès, and Zuowei Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2010.

[10] Ho Yee Cheung, Tsz Chiu Kwok, and Lap Chi Lau. Fast matrix rank algorithms and applications. In *Proceedings of the 44th Symposium on Theory of Computing (STOC)*, pages 549–562, 2012.

[11] Kenneth Clarkson, Petros Drineas, Malik Magdon-Ismail, Michael Mahoney, Xiangrui Meng, and David Woodruff. The fast Cauchy transform and faster robust linear regression. In *SODA*, pages 466–477, 2013.

[12] Kenneth L. Clarkson and David P. Woodruff. Numerical linear algebra in the streaming model. In *Proceedings of the 41st ACM Symposium on Theory of Computing (STOC)*, pages 205–214, 2009.

[13] Kenneth L. Clarkson and David P. Woodruff. Low rank approximation and regression in input sparsity time. In *Proceedings of the 45th ACM Symposium on Theory of Computing (STOC), (see also full version CoRR abs/1207.6365v3)*, 2013.

[14] Anirban Dasgupta, Petros Drineas, Boulos Harb, Ravi Kumar, and Michael W. Mahoney. Sampling algorithms and coresets for ℓ_p regression. *SIAM J. Comput.*, 38(5):2060–2078, 2009.

[15] Anirban Dasgupta, Ravi Kumar, and Tamás Sarlós. A sparse Johnson-Lindenstrauss transform. In *Proceedings of the 42nd ACM Symposium on Theory of Computing (STOC)*, pages 341–350, 2010.

[16] James Demmel, Ioana Dumitriu, and Olga Holtz. Fast linear algebra is stable. *Numer. Math.*, 108(1):59–91, October 2007.

[17] Petros Drineas, Malik Magdon-Ismail, Michael Mahoney, and David Woodruff. Fast approximation of matrix coherence and statistical leverage. In *Proceedings of the 29th International Conference on Machine Learning (ICML)*, 2012.

[18] Zoltán Füredi and János Komlós. The eigenvalues of random symmetric matrices. *Combinatorica*, 1(3):233–241, 1981.

[19] Yehoram Gordon. On Milman's inequality and random subspaces which escape through a mesh in $\mathbb{R}^n$. *Geometric Aspects of Functional Analysis*, pages 84–106, 1988.

[20] Nathan Halko, Per-Gunnar Martinsson, and Joel A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Rev., Survey and Review section*, 53(2):217–288, 2011.

[21] David Lee Hanson and Farroll Tim Wright. A bound on tail probabilities for quadratic forms in independent random variables. *Ann. Math. Statist.*, 42(3):1079–1083, 1971.

[22] Nicholas J. A. Harvey. *Matchings, Matroids and Submodular Functions.* PhD thesis, Massachusetts Institute of Technology, 2008.

[23] Aicke Hinrichs and Jan Vybíral. Johnson-Lindenstrauss lemma for circulant matrices. *Random Struct. Algorithms*, 39(3):391–398, 2011.

[24] William B. Johnson and Joram Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. *Contemporary Mathematics*, 26:189–206, 1984.

[25] Daniel M. Kane and Jelani Nelson. A derandomized sparse Johnson-Lindenstrauss transform. *CoRR*, abs/1006.3585, 2010.

[26] Daniel M. Kane and Jelani Nelson. Sparser Johnson-Lindenstrauss transforms. In *SODA*, pages 1195–1206, 2012.

[27] Daniel M. Kane, Jelani Nelson, Ely Porat, and David P. Woodruff. Fast moment estimation in data streams in optimal space. In *Proceedings of the 43rd ACM Symposium on Theory of Computing (STOC)*, pages 745–754, 2011.

[28] Oleksiy Khorunzhiy. Sparse random matrices: spectral edge and statistics of rooted trees. *Adv. Appl. Prob.*, 33:124–140, 2001.

[29] Oleksiy Khorunzhiy. Rooted trees and moments of large sparse random matrices. *Disc. Math. and Theor. Comp. Sci.*, AC:145–154, 2003.

[30] Bo'az Klartag and Shahar Mendelson. Empirical processes and random projections. *J. Funct. Anal.*, 225(1):229–245, 2005.

[31] Felix Krahmer, Shahar Mendelson, and Holger Rauhut. Suprema of chaos processes and the restricted isometry property. *Comm. Pure Appl. Math.*, to appear.

[32] 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.

[33] Michael W. Mahoney. Randomized algorithms for matrices and data. *Foundations and Trends in Machine Learning*, 3(2):123–224, 2011.

[34] Xiangrui Meng and Michael W. Mahoney. Low-distortion subspace embeddings in input-sparsity time and applications to robust linear regression. In *Proceedings of the 45th ACM Symposium on Theory of Computing (STOC), (see also full version CoRR abs/1210.3135*, 2013.

[35] Gary L. Miller and Richard Peng. Iterative approaches to row sampling. *CoRR*, abs/1211.2713, 2012.

[36] Crispin St. John Alvah Nash-Williams. Edge-disjoint spanning trees of finite graphs. *J. London Math. Soc.*, 36:445–450, 1961.

[37] Jelani Nelson and Huy L. Nguyễn. Sparsity lower bounds for dimensionality-reducing maps. In *Proceedings of the 45th ACM Symposium on Theory of Computing (STOC), to appear*, 2013.

[38] Nam H. Nguyen, Thong T. Do, and Trac D. Tran. A fast and efficient algorithm for low-rank approximation of a matrix. In *Proceedings of the 41st ACM Symposium on Theory of Computing (STOC)*, pages 215–224, 2009.

[39] Holger Rauhut. Compressive sensing and structured random matrices. In Massimo Fornasier, editor, *Theoretical Foundations and Numerical Methods for Sparse Recovery*, pages 1–92. De Gruyter, 2010.

[40] Tamás Sarlós. Improved approximation algorithms for large matrices via random projections. In *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 143–152, 2006.

[41] Arnold Schönhage. Unitäre transformationen großer matrizen. *Numer. Math.*, 20:409–417, 1973.

[42] Terence Tao. *Topics in random matrix theory*, volume 132 of *Graduate Studies in Mathematics*. American Mathematical Society, 2012.

[43] Mikkel Thorup and Yin Zhang. Tabulation-based 5-independent hashing with applications to linear probing and second moment estimation. *SIAM J. Comput.*, 41(2):293–331, 2012.

[44] Joel A. Tropp. Improved analysis of the subsampled randomized Hadamard transform. *Adv. Adapt. Data Anal., Special Issue on Sparse Representation of Data and Images*, 3(1–2):115–126, 2011.

[45] William Thomas Tutte. On the problem of decomposing a graph into n connected factors. *J. London Math. Soc.*, 142:221–230, 1961.

[46] Roman Vershynin. Introduction to the non-asymptotic analysis of random matrices. In Y. Eldar and G. Kutyniok, editors, *Compressed Sensing, Theory and Applications*, chapter 5, pages 210–268. Cambridge University Press, 2012.

[47] Jan Vybíral. A variant of the Johnson-Lindenstrauss lemma for circulant matrices. *J. Funct. Anal.*, 260(4):1096–1105, 2011.

[48] Eugene P. Wigner. Characteristic vectors of bordered matrices with infinite dimensions. *Ann. Math.*, 62:548–564, 1955.

[49] Virginia Vassilevska Williams. Multiplying matrices faster than Coppersmith-Winograd. In *STOC*, pages 887–898, 2012.

[50] Phillip Matchett Wood. Universality and the circular law for sparse random matrices. *Ann. Appl. Prob.*, 22(3):1266–1300, 2012.

[51] Yunhong Zhou, Dennis M. Wilkinson, Robert Schreiber, and Rong Pan. Large-scale parallel collaborative filtering for the Netflix prize. In *Proceedings of the 4th International Conference on Algorithmic Aspects in Information and Management (AAIM)*, pages 337–348, 2008.