Schatten Norms in Matrix Streams: Hello Sparsity, Goodbye Dimension

Vladimir Braverman[*] Robert Krauthgamer[†] Aditya Krishnan[‡] Roi Sinoff[†]

June 22, 2020

Abstract

Spectral functions of large matrices contains important structural information about the underlying data, and is thus becoming increasingly important. Many times, large matrices representing real-world data are *sparse* or *doubly sparse* (i.e., sparse in both rows and columns), and are accessed as a *stream* of updates, typically organized in *row-order*. In this setting, where space (memory) is the limiting resource, all known algorithms require space that is polynomial in the dimension of the matrix, even for sparse matrices. We address this challenge by providing the first algorithms whose space requirement is *independent of the matrix dimension*, assuming the matrix is doubly-sparse and presented in row-order. Our algorithms approximate the Schatten p-norms, which we use in turn to approximate other spectral functions, such as logarithm of the determinant, trace of matrix inverse, and Estrada index. We validate these theoretical performance bounds by numerical experiments on real-world matrices representing social networks. We further prove that multiple passes are unavoidable in this setting, and show extensions of our primary technique, including a trade-off between space requirements and number of passes.

1 Introduction

Large matrices are often used to represent real-world data sets like text documents, images and social networks, however analyzing them is increasingly challenging, as their sheer size renders many algorithms impractical. Fortunately, in several application domains, input matrices are often very sparse, meaning that only a small fraction of their entries are non-zero. In fact, in applications related to natural language processing (e.g. [GvDCB13]), image recognition, medical imaging and computer vision (e.g. [GJP$^+$12, LZYQ15]), the matrices are often doubly sparse, i.e., sparse in both rows and columns. Throughout, we define these matrices as *k-sparse*, meaning that every row and every column has at most k non-zero entries. The current work devises new algorithms to analyze the *spectrum* (singular values) of such sparse matrices, aiming to achieve efficiency (storage requirement in streaming model) that depends on *matrix sparsity* instead of *matrix dimension*.

We focus on fundamental functions of the spectrum, called the Schatten norms. Formally, the Schatten p-norm of a matrix $A \in \mathbb{R}^{m\times n}, m \geq n$ with singular values $\sigma_1 \geq \ldots \geq \sigma_n \geq 0$ is defined

[*] Johns Hopkins University. This research was supported in part by NSF CAREER grant 1652257, ONR Award N00014-18-1-2364 and and the Lifelong Learning Machines program from DARPA/MTO. Email: `vova@cs.jhu.edu`

[†] Weizmann Institute of Science. Work partially supported by ONR Award N00014-18-1-2364, the Israel Science Foundation grant #1086/18, and a Minerva Foundation grant. Part of this work was done while the author was visiting the Simons Institute for the Theory of Computing. Email: `{robert.krauthgamer,roi.sinoff}@weizmann.ac.il`

[‡] Johns Hopkins University. Work partially supported by ONR Award N00014-18-1-2364. Email: `akrish23@jhu.edu`

for every $p \geq 1$ as

$$\|A\|_{S_p} \coloneqq \left(\sum_{i=1}^{n} \sigma_i^p\right)^{1/p}.$$

This definition extends also to $0 < p < 1$, in which case it is not a norm, and also to $p = 0, \infty$ by taking the limit. Frequently used cases include $p = 0$, representing the rank of A, and $p = 1, 2, \infty$, commonly known as the trace/nuclear norm, Frobenius norm, and spectral/operator norm, respectively. Schatten norms are often used as surrogates for the spectrum, as explained in [ZWJ15, KV16, NPS16, KO19], and some specific cases have applications in optimization, image processing, and differential privacy etc. [XGL+16, MNS+18].

For a positive semidefinite (PSD) matrix A, the Schatten norms can be easily used to approximate other important spectral functions. One example is the Estrada index, which has applications in chemistry, physics, network theory and information theory (see survey by Gutman et al. [GDR11]). Another example is the trace of matrix inverse, which is used for image restoration, counting triangles in graphs, to measure the uncertainty in data collections, and to bound the total variance of unbiased estimators (see e.g. [WLK+16, Che16, HMAS17] for references). A third example is the logarithm of the determinant, used in many machine learning tasks, like Bayesian learning, kernel learning, Gaussian processes, tree mixture models, spatial statistics and Markov field models (see e.g. [HMS15, UCS17, US17, HMAS17] for references). Thus, our results for Schatten norm have further applications.

As the matrices in many real-world applications are often very large, storing the entire matrices in working memory can be impractical, and thus, as mentioned, analyzing them has become increasingly challenging. As a result, the data-stream model has emerged as a convenient model for how these data-sets are accessed in practice. In this model, the input matrix $A \in \mathbb{R}^{m\times n}$ is presented as a sequence of items/updates. In one common setting, the *turnstile* model, each update has the form (i, j, δ) for $\delta \in \mathbb{Z}$ and represents an update $A_{ij} \leftarrow A_{ij} + \delta$. In another common setting, the *row-order* model, items (i, j, A_{ij}) arrive in a fixed order, sorted by location (i, j) lexicographically, providing directly the entry of A in that location. In both models, unspecified entries are 0 by convention, which is very effective for sparse matrices.

Row-order is a common access pattern for external memory algorithms. When the data is too large to fit into working memory and has be "streamed" into memory in some pattern, it is useful to assume that algorithms can make multiple, albeit few, passes over the input data. For a thorough discussion of such external memory algorithms, including motivation for the row-order model and for multiple passes over the data, see [GM99, Vit01, Lib13].

Designing small-space algorithms for estimating Schatten norms of an input matrix in the data-stream model is an important problem, and was investigated recently for various matrix classes and stream types [CW09, AN13, LNW14, LW16a, LW16b, LW17, BCK+18]. However, all known algorithms require space that is polynomial in n, the matrix dimension, even if the matrix is highly sparse and the stream type is favorable, say row-order. A natural question then is:

> Does any streaming model admit algorithms for computing Schatten norms of a matrix presented as a stream, with storage requirement independent of the matrix dimension?

We *answer this question in the affirmative* for k-sparse matrices presented in row-order and all even integers p. Our algorithms extend to all integers $p \geq 1$ if the input matrix is PSD.

1.1 Main Results

Throughout, we write $\tilde{O}(f)$ as a shorthand for $O(f \cdot \log^{O(1)} n)$ where n is the dimension of the matrix, and write $O_d(f)$ when the hidden constant might depend on the parameter d. We assume

Table 1: Bounds for Schatten norms (for even p) of k-sparse matrices in row-order streams. Upper bound space is counted in words. Lower bounds are for suitable $\varepsilon(p) > 0$.

Which p	**Space Bound**	**Ref.**	**Comments**
$p > 4$	$\tilde{O}_{p,\varepsilon}(k^{O(p)} n^{1-4/\lceil p \rceil_4})$	[BCK+18]	one-pass
	$O_{p,\varepsilon}(k^{3p/2-3})$	Thm. 1.1	$\lfloor p/4 \rfloor + 1$ passes
	$O_{p,\varepsilon}(k^{2ps} n^{1-1/s})$	Thm. 5.3	$\lfloor \frac{p}{2(s+1)} \rfloor + 1$ passes
	$\Omega(n^{1-4/\lfloor p \rfloor_4})$	Thm. 1.2	one-pass, $k = O(1)$
	$\Omega_t(k^{p/2-2})$	[BCK+18]	t passes, $k \le n^{2/p}$
$p = 4$	$\tilde{O}_{p,\varepsilon}(k)$	[BCK+18]	one-pass
	$O_p(\varepsilon^{-2})$	Thm. 7.2	one-pass, for all $k \le n$

that the entries of the matrix are integers bounded by $\mathrm{poly}(n)$, and thus often count space in words, each having $O(\log n)$ bits. We denote by $\lceil p \rceil_4$ the smallest multiple of 4 that is greater than or equal to p, and similarly by $\lfloor p \rfloor_4$ the largest multiple of 4 that is smaller than or equal to p.

Upper and Lower Bounds for Row-Order Streams. Our main result is a new algorithm for approximating the Schatten p-norm (for even p) of a k-sparse matrix streamed in row-order, using $O(p)$ passes and $\mathrm{poly}(k^p/\varepsilon)$ space (independent of the matrix dimensions). This is stated in the next theorem, whose proof appears in Section 4.1.

Theorem 1.1. *There exists an algorithm that, given $p \in 2\mathbb{Z}_{\ge 2}$, $\varepsilon > 0$ and a k-sparse matrix $A \in \mathbb{R}^{n \times n}$ streamed in row-order, makes $\lfloor p/4 \rfloor + 1$ passes over the stream using $O_p(\varepsilon^{-2} k^{3p/2-3})$ words of space, and outputs $\bar{Y}(A)$ that $(1 \pm \varepsilon)$-approximates $\|A\|_{S_p}^p$ with probability at least $2/3$.*

Theorem 1.1 provides a multi-pass algorithm whose space complexity depends only on the sparsity of the input matrix. A natural question is whether one can achieve a similar dependence also for *one-pass* algorithms, but our next theorem (proved in Section 6) shows that such algorithms require $\Omega((n^{1-4/\lfloor p \rfloor_4})$ bits of space, even for $O(1)$-sparse matrices.

It follows that multiple passes over the data are necessary for an algorithm for sparse matrices to have space complexity independent of the matrix dimensions.

Theorem 1.2. *For every $p \in 2\mathbb{Z}_{\ge 2}$ there is $\varepsilon(p) > 0$ such that every algorithm that makes one pass over an $O_p(1)$-sparse matrix $A \in \mathbb{R}^{n \times n}$ streamed in row-order, and then outputs a $(1 \pm \varepsilon(p))$-approximation to $\|A\|_{S_p}^p$ with probability at least $2/3$, must use $\Omega(n^{1-4/\lfloor p \rfloor_4})$ bits of space.*

We can further extend our primary algorithmic technique (from Theorem 1.1) in several different ways, and obtain improved algorithms for special families of matrices, algorithms in the more general turnstile model, and algorithms with a trade-off between the number of passes and the space requirement, as explaind later in this section. Table 1 summarizes our results for row-order streams, and compares them to bounds derived from previous work (when applicable).

Applications for Approximating Schatten Norms. We show in Section 8 two settings where, under certain simplifying conditions, our algorithms can be used to approximate other functions of the spectrum, and even weakly recover the entire spectrum. The basic idea is that it suffices to compute only a few Schatten norms, in which case our algorithms for k-sparse matrices in row-order streams can be used, and the overall algorithm will require only small space (depending on k).

The first setting considers spectral sums of PSD matrices. We use an idea from [BDK$^+$17] to show that for a PSD input $A \in \mathbb{R}^{n\times n}$ whose eigenvalues lie in an interval $[\theta, 1)$, one can $(1 \pm 2\varepsilon)$-approximate $\log\det(A)$ and $\mathsf{Tr}(A^{-1})$ using the first $\frac{1}{\theta}\log\left(\frac{1}{\varepsilon}\right)$ (integer) Schatten norms. We further show that given a Laplacian matrix whose eigenvalues lie in an interval $[0, \theta]$, one can $(1 \pm 2\varepsilon)$-approximate the Estrada index using the first $(e\theta + 1)\log\frac{1}{\varepsilon}$ (integer) Schatten norms.

The second setting considers recovering the spectrum of a PSD matrix using a few Schatten norms of the matrix. We use an idea from [KV16] to approximate the spectrum of a PSD matrix whose eigenvalues lie in the interval $[0, 1]$, up to L_1-distance εn using the first $O(1/\varepsilon)$ Schatten norms.

Experiments. We validated our row-order algorithm on real-world matrices representing academic collaboration network graphs. The experiments show that the space needed to approximate the Schatten 6-norm of these matrices is much smaller than the theoretical bound, and that the algorithm is efficient also for larger p values. In fact, the matrices in our experiments have $O(1)$-sparse in every row, but their columns are only sparse on average. We also experimented to check if the algorithm is robust to noise, and found that it is indeed effective also for nearly-sparse matrices. Our experiments validate that the storage requirement is independent of the matrix dimensions. See Section 9 for details.

1.2 Extensions of Main Results

Extension I: Fewer Passes. We show in Section 5 how to generalize our algorithmic technique to use fewer passes over the stream, albeit requiring more space. Our method attains the following pass-space trade-off. For any integer $s \geq 2$, our algorithm in Theorem 5.3 makes $t(s) = \lfloor \frac{p}{2(s+1)} \rfloor + 1$ passes over the stream using $O_p\left(\varepsilon^{-3}k^{2ps}n^{1-1/s}\right)$ words of space, and outputs a $(1\pm\varepsilon)$-approximation to $\|A\|_{S_p}^p$ for $p \in 2\mathbb{Z}_{\geq 2}$.

Extension II: Turnstile Streams. We design in Section 4.2 an algorithm for *turnstile* streams with an additional $\tilde{O}(\varepsilon^{-O(p)}k^{3p/2-3}n^{1-2/p})$ factor in their space complexity compared to our algorithm for row-order streams. An additional $O(n^{1-2/p})$ factor is to be expected since the space complexity for estimating ℓ_p norms of vectors in turnstile streams is $\Omega(\frac{n^{1-2/p}}{t})$ if the algorithm is allowed to make t passes over the data. Our algorithm for turnstile streams makes $p+1$ passes over the stream. The algorithm of [LW16a] for $O(1)$-sparse matrices in the turnstile model can obviously be extended to k-sparse matrices. Its space requirement is $k^{O(p)}$, and we believe that a straightforward extension of their analysis yields an exponent greater than $4.75p$

Extension III: Special Matrix Families. We give in Section 4.1 improved bounds for special families of k-sparse matrices that may be of potential interest. We show that for Laplacians of undirected graphs with degree at most $k \in \mathbb{N}$, one can $(1 \pm \varepsilon)$-approximate the Schatten p-norm with space $O_p(\varepsilon^{-2}k^{p/2-1})$ by making $p/2$ passes over a row-order stream. Additionally, for matrices whose non-zero entries lie in an interval $[\alpha, \beta]$ for $\alpha, \beta \in \mathbb{R}^+$, we can get nearly-tight upper bounds – our algorithm uses space $O_p(\varepsilon^{-2}k^{p/2-1}(\beta/\alpha)^{p/2-2})$, which is nearly tight compared to the $\Omega(k^{p/2-2})$ multi-pass lower bound given in [BCK$^+$18] where $\alpha = \beta = 1$.

Schatten 4-norm. We show in Section 7 a simple one-pass algorithm for $(1 \pm \varepsilon)$-approximating the Schatten 4-norm of *any* matrix (not necessarily sparse) given in a row-order stream, using only $\tilde{O}_p(\varepsilon^{-2})$ words of space. This improves a previous $\tilde{O}_p(\varepsilon^{-2}k)$ bound from [BCK+18].

1.3 Technical Overview

Upper Bounds. We design an estimator that is inspired by the importance sampling framework and uses multiple passes over the data to implement the estimator. To the best of the our knowledge, this is the first algorithm for computing the Schatten p-norm in data streams that uses an adaptive sampling approach, i.e. the probability space of the algorithm's sampling in a given pass of the data is affected by the algorithm's decisions in the previous pass.

For an integer $p \in 2\mathbb{Z}_{\geq 1}$ and $q := p/2$, the Schatten p-norm for a matrix $A \in \mathbb{R}^{n\times n}$, denoted by $\|A\|_{S_p}^p$, can be expressed as

$$\|A\|_{S_p}^p = \mathsf{Tr}((AA^\top)^q) = \sum_{i_1,\dots,i_q \in [n]} \langle a_{i_1}, a_{i_2}\rangle\langle a_{i_2}, a_{i_3}\rangle \dots \langle a_{i_q}, a_{i_1}\rangle \tag{1.1}$$

where a_i is the i^{th} row of matrix A.

The Schatten p-norm can be interpreted using (1.1) as a sum over cycles of q inner-products (which we refer to informally as *cycles*) between rows of A. We assign each cycle in the above expression to one of the rows participating in that cycle. Hence, the Schatten p-norm can be expressed as a sum $\sum_{i=1}^n z_i$ where z_i is the cumulative weight of all the cycles assigned to row i.

Our algorithm starts by sampling a row $i \in [n]$ with probability proportional to the *heaviest* cycle assigned to row i (i.e., largest contribution to z_i). In the following $p/4$ stages, it samples one cycle assigned to i with probability proportional to the weight of the cycle. Since the rows *and* columns are sparse, each row cannot participate in "too many" cycles (because it is orthogonal to any row with a disjoint support). Specifically, we show that the number of cycles assigned to each row i is only a function of k and p. It follows that sampling the first row with probability proportional to the heaviest contributing cycle is a good approximation (up to a factor depending only on k and p) to z_i, the actual contribution of row i to $\sum_{i\in[n]} z_i = \|A\|_{S_p}^p$.

The space complexity of sampling a row with probability proportional to its heaviest contributing cycle depends on the assigning process. A natural assigning is to assign every cycle to the row with largest l_2-norm participating in that cycle (breaking ties arbitrarily). Notice then that, by the Cauchy-Schwarz inequality, the heaviest contributing cycle to row i is simply $\|a_i\|_2^p$.

This estimator can be implemented in the row-order model easily by using weighted reservoir sampling [Vit85, BOV15], as shown in Section 4.1. However, implementing it in turnstile streams is more challenging (see Section 4.2). Using approximate L_p-samplers presented in [MW10], we build an approximate cascaded $L_{p,2}$-norm[1] sampler, to sample rows i with probability proportional to $\|a_i\|_2^p$. Additionally, we use the Count-Sketch data structure to recover rows and sample cycles once we have sampled the first, "seed" row. This allows us to implement the estimator in turnstile data streams with an additional $\tilde{O}(\varepsilon^{-O(p)}n^{1-2/p})$ factor in the space complexity attributed to the approximate cascaded $L_{p,2}$-norm sampler and an additional $O_p(k^{3p/2-3})$ factor that comes from approximating the sampling probabilities (compared to the row-order in which the sampling probabilities can be recovered exactly).

In Section 5 we generalize the design of the importance sampling estimator. Instead of assigning every cycle to a single row that appears in it, every cycle is mapped to s rows that participate in it, for parameter $s \in \mathbb{N}$. These s rows split the cycle into roughly $\frac{q}{s}$ segments such that each of

[1] The $L_{p,2}$-norm of a matrix $A \in \mathbb{R}^{n\times m}$ for $p \geq 0$ is $\left(\sum_{i=1}^n \|a_i\|_2^p\right)^{1/p}$.

these s rows participates in a segment where it is the heaviest' row (by l_2-norm). The algorithm samples s "seed" rows and then computes all the cycles (or alternatively samples one cycle) that are assigned to these s rows. Since the length of each of the segments reduces linearly with s, one can compute these cycles with fewer passes. However, the algorithm needs to sample more indices in order to ensure that each cycle has a sufficiently large probability of being "hit". This tension leads to a trade-off between passes and space.

Lower Bounds. We obtain an $\Omega(n^{1-4/\lfloor p \rfloor_4})$ bits lower bound for any algorithm that estimates the Schatten p-norm in one-pass of the stream for even p values. Our proof analyzes for even p values a construction presented in [LW16a], which is based on a reduction from the Boolean Hidden Hypermatching problem. This lower bound holds even if the input matrix is promised to be $O(1)$-sparse.

1.4 Previous and Related Work

The bilinear sketching algorithm in [LNW14] was the first non-trivial algorithm for Schatten p-norm estimation in turnstile streams. It requires only one-pass over the data and uses $O(\varepsilon^{-2} n^{2-4/p})$ words of space.[2] Their algorithm uses $O(\varepsilon^{-2})$ independent $G_1 A G_2^\top$ sketches, where $G_1, G_2 \in \mathbb{R}^{t\times n}$ are matrices with i.i.d. Gaussian entries and $t = O(n^{1-2/p})$.

Inspired by this sketch, [BCK+18] gave an almost quadratic improvement in the space complexity if the algorithm is allowed to make multiple passes over the data. Their estimator uses matrices $G_2, \dots, G_p \in \mathbb{R}^{t\times n}$ with i.i.d. Gaussian entries and Gaussian vector $g_1 \in \mathbb{R}^n$ to output $g_1^\top A G_2^\top G_2 A \dots G_p A g_1$. This estimate can be constructed in $p/2$ passes of the data and requires $O(\varepsilon^{-2})$ independent copies each using only $t = O(n^{1-\frac{1}{p-1}})$ space.

Restricting the input matrix to be $O(1)$-sparse allows for quadratic improvement in the space complexity of one-pass algorithms as shown in [LW16a]. They show that sampling $O(n^{1-2/p})$ rows and storing them approximately using small space (since each row is sparse) is sufficient to $(1+\varepsilon)$-approximate the Schatten p-norm by exploiting the fact that rows cannot "interact" with one another "too much" because of the sparsity restriction.

If we restrict the data stream to be row-order, then we can reduce the dependence on p in all the above algorithms by a factor of 2. As noted in [BCK+18], since $A^\top A = \sum_i a_i a_i^\top$ (where a_i is the i^{th} row of A) one can apply the above algorithms to $A^\top A$ instead of A by updating it with the outer product of every row with itself. Since $\|A^\top A\|_{S_{p/2}}^{p/2} = \|A\|_{S_p}^p$ (for even p values), the output is as desired and the dependence on p reduces by a factor of 2.

Lower Bounds. Every t-pass algorithm designed for turnstile streams requires $\Omega(n^{1-2/p}/t)$ bits, which follows by injecting the F_p-moment problem (see [Gro09, Jay09]) into the diagonal elements. Li and Woodruff [LW16a] showed that restricting the algorithm to a single pass over the turnstile stream, leads to a lower bound $\Omega(n^{1-\varepsilon})$ bits for every fixed $\varepsilon > 0$ and $p \notin 2\mathbb{Z}_{\geq 2}$, even if the input matrix is $O(1)$-sparse.[3] Later [BCK+18] proved that $\Omega(n^{1-\varepsilon})$ bits are required for $p \notin 2\mathbb{Z}_{\geq 2}$ even in row-order streams. In addition, they showed (Theorem 5.4 in Arxiv version) that t passes over row-order streams require space $\Omega(n^{1-4/p}/t)$ bits, however these matrices are actually $\Omega(n^{2/p})$-sparse

[2] They also showed a lower bound of $\Omega(n^{2-4/p})$ for the dimension of bilinear sketching for approximating $\|A\|_{S_p}^p$ for all $p \geq 2$.

[3] They also showed that for $p \in 2\mathbb{Z}_{\geq 2}$, single-pass algorithms require $\Omega(n^{1-2/p})$ bits even if all non-zeros in the input matrix are constants.

(and not $O(1)$-sparse as may be understood from Table 2 therein). A simple adaptation of that result yields an $\Omega(k^{p/2-2}/t)$ space lower bound for k-sparse input matrices ($k \le n^{2/p}$).

2 Notation and Preliminaries

The following useful fact comparing the lengths of the rows of A and its Schatten p-norm is proved in Appendix A.1.

Fact 2.1. *Let matrix $A \in \mathbb{R}^{n\times n}$ have rows $\{a_i\}_{i\in[n]}$ and let $t \ge 1$. Then $\sum_{i\in[n]} \|a_i\|_2^{2t} \le \|A\|_{S_{2t}}^{2t}$.*

Importance Sampling. Our main algorithmic technique is inspired by the importance sampling framework, as formulated by the following theorem, proved in Appendix A.2.

Theorem 2.2 (Importance Sampling)**.** *Let $z = \sum_{i\in[n]} z_i \ge 0$ be a sum of n reals. Let the random variable $\hat{Z}$ be an estimator computed by sampling a single index $i \in [n]$ according to the probability distribution given by $\{\tau_i\}_{i=1}^n$ and setting $\hat{Z} \leftarrow \frac{z_i}{\tau_i}$. If for some parameter $\lambda \ge 1$, each $\tau_i \ge \frac{|z_i|}{\lambda\cdot z}$, then*

$$\mathbb{E}\left[\hat{Z}\right] = z \quad \textit{and} \quad \mathrm{Var}(\hat{Z}) \le (\lambda z)^2.$$

Families of Matrices. We define two families of matrices that are of special interest.

- Let $\mathcal{L}_n \subseteq \mathbb{Z}^{n\times n}$ be the family of Laplacian matrices of undirected graphs $G([n], E)$ with positive edge-weights $\{w_{uv} > 0 : uv \in E\}$.
- Given positive constants $\alpha \le \beta$, let $\mathcal{C}_{\alpha,\beta}^{m\times n} \subseteq \mathbb{R}^{m\times n}$ be the family of matrices C such that every entry $C_{i,j}$ is either zero or in the range $[\alpha, \beta]$. For the vector case (i.e. $n = 1$) we may write $\mathcal{C}_{\alpha,\beta}^{m}$.

3 An Estimator for Schatten p-Norm for $p \in 2\mathbb{Z}_{\ge 2}$

This section introduces our importance sampling estimator for Schatten p-norms. We begin in Section 3.1 with manipulating expression (1.1) for the Schatten p-norm by assigning every summand, i.e. a cycle of $p/2$ inner products, to its heaviest participating row, see (3.3). We then use this new expression in Section 3.2 to give an importance sampling estimator. In Section 3.3 we prove several lemmas, referred to as projection lemmas, which are key to our analysis in Section 3.4.

3.1 Preliminaries

Fix a matrix $A \in \mathbb{R}^{n\times n}$ and $p \in 2\mathbb{Z}_{\ge 2}$. For a row a_i, we define the set of its *neighboring rows* $N(i) \coloneqq \{l \in [n] : \mathrm{supp}(a_i) \cap \mathrm{supp}(a_l) \ne \emptyset\}$. In addition, we denote the set of neighboring rows of a_j that have smaller length than row a_i

$$N_S^i(j) \coloneqq \{l \in N(j) : \ \|a_l\|_2 \le \|a_i\|_2\}.$$

Building on this, we intordure notation for certain "paths" of rows. Fixing some row indices $i, i_1 \in [n]$ and an integer $t \ge 2$, we then define

$$\Gamma(i_1, t) \coloneqq \{(i_1, \dots, i_t) : \ i_2 \in N(i_1), \dots, i_t \in N(i_{t-1})\},$$
$$\Gamma_S^i(i_1, t) \coloneqq \left\{(i_1, \dots, i_t) : \ i_2 \in N_S^i(i_1), \dots, i_t \in N_S^i(i_{t-1})\right\}.$$

We further define the weights of "paths" of inner products: given an integer $t \geq 2$ and indices $i_1, \ldots, i_t \in [n]$, let

$$\sigma(i_1, \ldots, i_t) \coloneqq \langle a_{i_1}, a_{i_2} \rangle \langle a_{i_2}, a_{i_3} \rangle \ldots \langle a_{i_{t-1}}, a_{i_t} \rangle.$$

Recall from (1.1) that the Schatten p-norm of $A \in \mathbb{R}^{n \times n}$ can be expressed in terms of the product of inner products of the rows of A. Using the above notation we manipulate it as follows.

$$\begin{aligned} \|A\|_{S_p}^p = \mathsf{Tr}\left((AA^\top)^q\right) &= \sum_{i_1,\ldots,i_q \in [n]} \sigma(i_1, \ldots, i_q, i_1) && (3.1)\\ &= \sum_{i_1} \sum_{\substack{(i_1,\ldots,i_{q-1})\\ \in \Gamma(i_1,q-1)}} \sum_{i_q \in N(i_1)} \sigma(i_1, \ldots, i_q, i_1) && (3.2)\\ &= \sum_{i_1} \sum_{\substack{(i_1,\ldots,i_{q-1})\\ \in \Gamma_S^{i_1}(i_1,q-1)}} \sum_{i_q \in N_S^{i_1}(i_1)} c(i_1, \ldots, i_q) \sigma(i_1, \ldots, i_q, i_1) && (3.3) \end{aligned}$$

where $1 \leq c(i_1, \ldots, i_q) \leq q$ is the number of times the sequence $(i_1, \ldots, i_q, i_1)$ or a cyclic shift of the sequence appears in Equation (3.2).

3.2 The Estimator

Our estimator is an importance sampling estimator for the quantity in (3.3). To define it, we need the following quantities:

$$\begin{aligned} \mathcal{S} &\coloneqq \bigcup_{i \in [n]} \Gamma_S^i(i, q-1) \\ z_{(i_1,\ldots,i_{q-1})} &\coloneqq \sum_{i_q \in N_S^{i_1}(i_1)} c(i_1, \ldots, i_q) \sigma(i_1, \ldots, i_q) \langle a_{i_q}, a_{i_1} \rangle && \forall (i_1, \ldots, i_{q-1}) \in \mathcal{S} \\ z &\coloneqq \sum_{(i_1,\ldots,i_{q-1}) \in \mathcal{S}} z_{(i_1,\ldots,i_{q-1})} = \|A\|_{S_p}^p && \text{by Equation (3.3).} \end{aligned}$$

Our importance sampling estimator, for the sum z, samples quantities $z_{(i_1,\ldots,i_{q-1})}$ indexed by $(i_1, \ldots, i_{q-1}) \in \mathcal{S}$ in $q-1$ steps. In the first step, it samples row $i_1 \in [n]$ with probability $\frac{\|a_{i_1}\|_2^p}{\sum_j \|a_j\|_2^p}$. In each step $2 \leq t \leq q-1$, conditioned on sampling i_{t-1} in step $t-1$ it samples row $i_t \in N_S^{i_1}(i_{t-1})$ with probability

$$p_{i_{t-1}}^{i_1}(i_t) \coloneqq \frac{|\langle a_{i_{t-1}}, a_{i_t} \rangle|}{\sum_{l \in N_S^{i_1}(i_{t-1})} |\langle a_{i_{t-1}}, a_l \rangle|}.$$

Overall, a sequence $(i_1, \ldots, i_{q-1}) \in \mathcal{S}$ is sampled with probability

$$\tau_{(i_1,\ldots,i_{q-1})} = \frac{\|a_{i_1}\|_2^p}{\sum_j \|a_j\|_2^p} \prod_{t=2}^{q-1} p_{i_{t-1}}^{i_1}(i_t),$$

and the output estimator is

$$Y(A) \coloneqq \frac{1}{\tau_{(i_1,\ldots,i_{q-1})}} \cdot z_{(i_1,\ldots,i_{q-1})}.$$

3.3 Projection Lemmas

To analyze the estimator $Y(A)$, we need a few lemmas, which we call projection lemmas, for sparse matrices. We start with two lemmas for sparse matrices, followed by two lemmas for more specialized cases.

Lemma 3.1. *For every k-sparse matrix $B \in \mathbb{R}^{n \times k}$ with rows $b_1, \ldots, b_n$ and vector $x \in \mathbb{R}^k$ such that $\|x\|_2 \geq \|b_i\|_2$ for all $i \in [n]$, we have that*

$$\frac{\|Bx\|_1}{\|x\|_2^2} = \sum_{i=1}^{n} \frac{|\langle x, b_i \rangle|}{\|x\|_2^2} \leq k\sqrt{k}.$$

Proof. For a vector $y \in \mathbb{R}^k$ and $S \subseteq [k]$, let $y_{|S}$ to be the restriction of y onto its indices corresponding to set S.

For all $i \in [n]$, by the Cauchy-Schwarz inequality, $\langle x, b_i \rangle = \langle x_{|\operatorname{supp}(b_i)}, b_i \rangle \leq \|x_{|\operatorname{supp}(b_i)}\|_2 \|b_i\|_2$. Hence,

$$\sum_{i=1}^{n} \frac{|\langle x, b_i \rangle|}{\|x\|_2^2} \leq \sum_{i=1}^{n} \frac{\|x_{|\operatorname{supp}(b_i)}\|_2 \|b_i\|_2}{\|x\|_2^2} \leq \sum_{i=1}^{n} \frac{\|x_{|\operatorname{supp}(b_i)}\|_2}{\|x\|_2} \leq \sum_{i=1}^{n} \frac{\|x_{|\operatorname{supp}(b_i)}\|_1}{\|x\|_2} \leq \frac{k\|x\|_1}{\|x\|_2},$$

where the last inequality follows from the sparsity of B (every column index is in $\operatorname{supp}(b_i)$ for at most k of the rows b_i). The lemma now follows by a simple application of the Cauchy-Schwarz inequality. □

We need another, similar, lemma in order to bound the variance.

Lemma 3.2. *For every k-sparse matrix $B \in \mathbb{R}^{n \times k}$ with rows $b_1, \ldots, b_n$ and a vector $x \in \mathbb{R}^k$ such that $\|x\|_2 \geq \|b_i\|_2$ for all $i \in [n]$, we have that*

$$\frac{\|Bx\|_2^2}{\|x\|_2^4} = \sum_{i=1}^{n} \frac{\langle x, b_i \rangle^2}{\|x\|_2^4} \leq k.$$

Proof. Following similar steps as that of Lemma 3.1,

$$\sum_{i=1}^{n} \frac{\langle x, b_i \rangle^2}{\|x\|_2^4} \leq \sum_{i=1}^{n} \frac{\|x_{|\operatorname{supp}(b_i)}\|_2^2}{\|x\|_2^2} \leq k,$$

where again the last inequality follows from the sparsity of B. □

The next two lemmas present bounds that improve over Lemma 3.1 in two special cases, when the k-sparse matrix is a graph Laplacian, and when all its non-zero entries come from a bounded range.

Lemma 3.3. *Let $G = ([n], E)$ be an undirected graph with positive edge weights $\{w_{uv}\}_{uv \in E}$. Let k be its maximum (unweighted) degree, and let $L(G) \in \mathbb{R}^{n \times n}$ be its Laplacian matrix with rows $l_1, \ldots, l_n$. Given $u \in [n]$, let the matrix B_u consist of all the rows l_v where $\|l_u\|_2 \geq \|l_v\|_2$, and interpret B_u also as a set of rows. Then,*

$$\frac{\|B_u l_u\|_1}{\|l_u\|_2^2} = \sum_{l_v \in B_u} \frac{|\langle l_u, l_v \rangle|}{\|l_u\|_2^2} \leq 2k.$$

(Trivially, we can also omit from B_u rows where $\langle l_u, l_v \rangle = 0$.)

Proof. The main idea is that the additional matrix structure implies $\|l_u\|_1 \le 2\|l_u\|_2$, which is better than what follows from the Cauchy-Schwarz inequality. Indeed, $\|l_u\|_2^2 = \left(-\sum_{t\in N(u)} w_{ut}\right)^2 + \sum_{t\in N(u)} w_{ut}^2 \ge \left(\sum_{t\in N(u)} w_{ut}\right)^2 = \left(\frac{1}{2}\|l_u\|_1\right)^2$. Now using this inequality in the proof of Lemma 3.1, we have

$$\frac{\|B_u l_u\|_1}{\|l_u\|_2^2} \le \frac{k\|l_u\|_1}{\|l_u\|_2} \le 2k.$$

□

Lemma 3.4. *For positive constants $\alpha \le \beta$ and a k-sparse matrix $B \in \mathcal{C}_{\alpha,\beta}^{n\times k}$ with rows $b_1, \dots, b_n$ and a vector $x \in \mathcal{C}_{\alpha,\beta}^{k}$ such that $\|x\|_2 \ge \|b_i\|_2$ for all $i \in [n]$, we have that*

$$\frac{\|Bx\|_1}{\|x\|_2^2} = \sum_{i=1}^{n} \frac{|\langle x, b_i\rangle|}{\|x\|_2^2} \le k\frac{\beta}{\alpha}.$$

Proof. By a direct calculation using the sparsity of B,

$$\sum_{i=1}^{n} \frac{|\langle x, b_i\rangle|}{\|x\|_2^2} \le \sum_{j=1}^{k} \frac{|x_j|\cdot \beta k}{\alpha\|x\|_1} = k\frac{\beta}{\alpha}.$$

□

3.4 Analyzing the Estimator

We now prove that the importance sampling estimator $Y(A)$ given in Section 3.2 is an unbiased estimator with a small variance. In addition to analyzing the estimator for all k-sparse matrices, we provide in Theorem 3.6 improved bounds for two special families of k-sparse matrices: (i) Laplacians of undirected graphs and (ii) matrices whose non-zero entries lie in an interval $[\alpha, \beta]$ for parameters $0 < \alpha \le \beta$.

Theorem 3.5. *For every $p \in 2\mathbb{Z}_{\ge 2}$ and a k-sparse matrix $A \in \mathbb{R}^{n\times n}$, the estimator $Y(A)$ given in Section 3.2 satisfies $\mathbb{E}\left[Y(A)\right] = \|A\|_{S_p}^p$ and $\mathrm{Var}(Y(A)) \le O_p(k^{\frac{3p}{2}-4})\|A\|_{S_p}^{2p}$.*

Proof. We will use the importance sampling framework of Theorem 2.2. In order to do so we must first argue that the values $\tau_{(i_1,\dots,i_{q-1})}$ for $(i_1,\dots,i_{q-1}) \in \mathcal{S}$ indeed form a probability distribution. It is easy to see that the probabilities of sampling the first row form a distribution over $[n]$. Similarly, for every $2 \le t \le q-1$, the values $p_{i_{t-1}}^{i_1}(\cdot)$ indeed form a probability distribution over the rows in $N_S^{i_1}(i_{t-1})$. The argument for $\tau_{(i_1,\dots,i_{q-1})}$ follows by the law of total probability.

Per Theorem 2.2, it is sufficient to prove that for all $(i_1,\dots,i_{q-1}) \in \mathcal{S}$,

$$\frac{1}{\tau_{(i_1,\dots,i_{q-1})}} \cdot \left|z_{(i_1,\dots,i_{q-1})}\right| \le O_p(k^{\frac{3}{4}p-2})z \tag{3.4}$$

Fix a sequence of indices $(i_1, \ldots, i_{q-1}) \in \mathcal{S}$. Inequality (3.4) can be shown as follows,

$$\begin{aligned}
\frac{\left|z_{(i_1,\ldots,i_{q-1})}\right|}{\tau_{(i_1,\ldots,i_{q-1})}} &= \frac{\sum_j \|a_j\|_2^p}{\|a_{i_1}\|_2^p} \prod_{t=2}^{q-1} \frac{1}{p^{i_1}_{i_{t-1}}(i_t)} \left| \sum_{i_q \in N_S^{i_1}(i_1)} c(i_1,\ldots,i_q)\sigma(i_1,\ldots,i_q)\langle a_{i_q}, a_{i_1}\rangle \right| \\
&\le \frac{\sum_j \|a_j\|_2^p}{\|a_{i_1}\|_2^p} \frac{\prod_{t=2}^{q-1} \sum_{l \in N_S^{i_1}(i_{t-1})} |\langle a_{i_{t-1}}, a_l\rangle|}{|\sigma(i_1,\ldots,i_{q-1})|} \sum_{i_q \in N_S^{i_1}(i_1)} c(i_1,\ldots,i_q) \left|\sigma(i_1,\ldots,i_q)\langle a_{i_q}, a_{i_1}\rangle\right| \\
&= \frac{\sum_j \|a_j\|_2^p}{\|a_{i_1}\|_2^p} \left(\prod_{t=2}^{q-1} \sum_{l \in N_S^{i_1}(i_{t-1})} |\langle a_{i_{t-1}}, a_l\rangle| \right) \sum_{i_q \in N_S^{i_1}(i_1)} c(i_1,\ldots,i_q) \left|\langle a_{i_{q-1}}, a_{i_q}\rangle\langle a_{i_q}, a_{i_1}\rangle\right|
\end{aligned}$$

By Young's Inequality for products of numbers and the bound on $c(i_1, \ldots, i_q)$,

$$\begin{aligned}
&\le \frac{q}{2} \frac{\sum_j \|a_j\|_2^p}{\|a_{i_1}\|_2^p} \left(\prod_{t=2}^{q-1} \sum_{l \in N_S^{i_1}(i_{t-1})} |\langle a_{i_{t-1}}, a_l\rangle| \right) \left(\sum_{i_q \in N_S^{i_1}(i_1)} \langle a_{i_{q-1}}, a_{i_q}\rangle^2 + \langle a_{i_q}, a_{i_1}\rangle^2 \right) \\
&= \frac{q}{2} \sum_j \|a_j\|_2^p \left(\frac{\prod_{t=2}^{q-1} \sum_{l \in N_S^{i_1}(i_{t-1})} |\langle a_{i_{t-1}}, a_l\rangle|}{\|a_{i_1}\|_2^{p-4}} \right) \left(\sum_{i_q \in N_S^{i_1}(i_1)} \frac{\langle a_{i_{q-1}}, a_{i_q}\rangle^2 + \langle a_{i_q}, a_{i_1}\rangle^2}{\|a_{i_1}\|_2^4} \right)
\end{aligned}$$

By applying Lemma 3.2 to the two inner-most summations and the fact that $\|a_{i_{q-1}}\|_2 \le \|a_{i_1}\|_2$,

$$\le qk \cdot \sum_j \|a_j\|_2^p \left(\frac{\prod_{t=2}^{q-1} \sum_{l \in N_S^{i_1}(i_{t-1})} |\langle a_{i_{t-1}}, a_l\rangle|}{\|a_{i_1}\|_2^{p-4}} \right)$$

By applying Lemma 3.1 and the fact that $\|a_{i_{t-1}}\|_2 \le \|a_{i_1}\|_2$ for any $2 \le t \le q-1$,

$$\le qk \sum_j \|a_j\|_2^p \left(\prod_{t=2}^{q-1} k\sqrt{k} \right) = qk^{\frac{3p}{4}-2} \sum_i \|a_i\|_2^p \le \frac{pk^{\frac{3p}{4}-2}}{2} \|A\|_{S_p}^p$$

where the last inequality follows from Fact 2.1. □

Theorem 3.6. *For every $p \in 2\mathbb{Z}_{\ge 2}$ and a k-sparse Laplacian matrix $A \in \mathcal{L}_n$, the estimator $Y(A)$ given in Section 3.2 satisfies* $\mathrm{Var}(Y(A)) \le O_p(k^{p/2-1})\|A\|_{S_p}^{2p}$. *If instead the k-sparse matrix is $A \in \mathcal{C}_{\alpha,\beta}^{n\times n}$ for some $0 < \alpha \le \beta$, then* $\mathrm{Var}(Y(A)) \le O_p(k^{p/2-2}\,(\beta/\alpha)^{p/2-2})\|A\|_{S_p}^{2p}$.

Proof. The bound for $\mathcal{L}_n$ (Laplacians) follows the above proof of Theorem 3.5 but bounding the summations using Lemma 3.3 instead of Lemma 3.1.

The bound for $\mathcal{C}_{\alpha,\beta}^{n\times n}$ uses a special case of the importance sampling lemma. Using the notation from Theorem 2.2, if $z_i > 0$ for all $i \in [n]$ then one can bound the variance by $\lambda(z)^2$. Using this, the proof follows the same argument as that of Theorem 3.5 but using Lemma 3.4 to bound the summations bounded by Lemmas 3.2 and 3.1. □

4 Implementing the Estimator: Row-Order and Turnstile Streams

In this section we show how to implement the importance sampling estimator defined in Section 3.2 in two different streaming models, row-order and turnstile streams. We start by stating two theorems that bound the space complexity of implementing the estimator in row-order streams. The first one is our main result from the Introduction, and applies to all k-sparse matrices. The second theorem considers special families of k-sparse matrices.

Theorem 1.1. *There exists an algorithm that, given $p \in 2\mathbb{Z}_{\geq 2}$, $\varepsilon > 0$ and a k-sparse matrix $A \in \mathbb{R}^{n \times n}$ streamed in row-order, makes $\lfloor p/4 \rfloor + 1$ passes over the stream using $O_p(\varepsilon^{-2} k^{3p/2-3})$ words of space, and outputs $\bar{Y}(A)$ that $(1 \pm \varepsilon)$-approximates $\|A\|_{S_p}^p$ with probability at least $2/3$.*

Theorem 4.1. *There exists an algorithm that, given $p \in 2\mathbb{Z}_{\geq 2}$, $\varepsilon > 0$, and a k-sparse matrix $A \in \mathcal{L}_n$ streamed in row-order, makes $\lfloor p/4 \rfloor + 1$ passes over the stream using $O_p(\varepsilon^{-2} k^{p/2})$ words of space, and outputs $\bar{Y}(A)$ that $(1 \pm \varepsilon)$-approximates $\|A\|_{S_p}^p$ with probability at least $2/3$. If instead the k-sparse matrix A is from $\mathcal{C}_{\alpha,\beta}^{n \times n}$ for $0 < \alpha \leq \beta$, then the space bound is $O_p(\varepsilon^{-2} k^{p/2-1} (\beta/\alpha)^{p/2-2})$ words.*

We also show that the estimator defined in Section 3.2 can be implemented in turnstile streams in $p/2 + 3$ passes over the stream.

Theorem 4.2. *There exists an algorithm that, given $p \in 2\mathbb{Z}_{\geq 2}$, $\varepsilon > 0$ and a k-sparse matrix $A \in \mathbb{R}^{n \times n}$ streamed in a turnstile fashion, makes $p/2 + 3$ passes over the stream using $O_p(k^{3p-6} n^{1-\frac{2}{p}} (\varepsilon^{-1} \log n)^{O(p)})$ words of space, and outputs $\bar{Y}(A)$ that $(1 \pm \varepsilon)$-approximates $\|A\|_{S_p}^p$ with probability at least $2/3$.*

Outline. At a high level, the algorithms in all three theorems are similar, and compute multiple copies of the estimator defined in Section 3.2 in parallel and output their average (to reduce the variance). The algorithms differ in the number of copies, derived from Theorems 3.5 and 3.6. Here, and in Sections 4.1 and 4.2, we describe how to implement each estimator in $p/2$ stages, and in Section 4.3 we show how to reduce the number of stages to $\lfloor p/4 \rfloor + 1$. The first stage samples and stores a "seed" row which we will denote by a_{i_1}. Each subsequent stage $1 < t < q$ stores two values: a row index i_t (and row a_{i_t} itself) and an interim estimate $Y_t \coloneqq \sigma(i_1, \ldots, i_t)$. The final stage q computes and outputs $\sum_{i_q \in N_S^{i_1}(i_1)} Y_{q-1} \cdot \langle a_{i_q}, a_{i_1} \rangle c(i_1, \ldots, i_q)$, where $1 \leq c(i_1, \ldots, i_q) \leq q$ is as defined in (3.3).

The estimator is relatively easy to implement in row-order streams using $p/2$ passes and $O_p(\varepsilon^{-2} k^{3p/2-3})$ words of space as shown in Section 4.1. In turnstile streams however, the estimator is more difficult to implement. The technical roadblock is sampling the first, "seed" row $i_1 \in [n]$ with probability proportional to $\frac{\|a_{i_1}\|_2^p}{\sum_j \|a_j\|_2^p}$. We use approximate samplers for turnstile streams to get around this roadblock. For a vector $x \in \mathbb{R}^n$ updated in a turnstile fashion, one can sample an index i with probability approximately $x_i^t / \|x\|_t^t$ for various $t \in [0, \infty)$. Such algorithms are called L_t-samplers and have been studied thoroughly, see e.g. [CJ19]. Approximate samplers introduce a multiplicative (relative) error and an additive error in the sampling probability, which need to be accounted for when analyzing the algorithm that uses the sampler.

Thus, in order to sample rows proportional to the quantities we want, we build two subroutines in the turnstile model:

1. Cascaded $L_{p,2}$-norm sampler for A, used to sample the seed row i_1 with probability approximately $\|a_{i_1}\|_2^p$. It runs in 2-*passes*, has relative error $O(\varepsilon)$ and uses space $\tilde{O}_p(\varepsilon^{-2} n^{1-2/p})$.

2. Compute inner products between a given row and its neighbors in space $\tilde{O}(k^2)$.

Using the two subroutines we can implement the estimator in Section 3.2 in $p+1$ passes of the stream in space $O_p(k^{3p-6}n^{1-2/p}(\varepsilon^{-1}\log n)^{O(p)})$. The additional $\tilde{O}(n^{1-2/p})$ space complexity factor is introduced by the approximate $L_{p,2}$-sampler. We remark that this factor is actually unavoidable for algorithms that compute $\|A\|_{S_p}^p$ in the turnstile model, since there is an $\Omega(n^{1-2/p})$ lower bound for computing the l_p-norm of vectors in $\mathbb{R}^n$ (in turnstile streams), even if the algorithm is allowed multiple passes. The additional $O(k^{3p/2-3})$ factor in the space complexity for turnstile streams compared to row-order streams is due to the bias introduced in estimating the sampling probability of the first, "seed" row.

As mentioned earlier, a slightly improved version runs in $\lfloor p/4 \rfloor + 1$ and $p/2+3$ passes for row-order and turnstile streams respectively, with the same space complexities (up to constant factors). The idea is to build two parallel paths from the same seed row and eventually "stitch" the two into one cycle.

4.1 Row-Order Streams

In this section we show how to easily implement the estimator defined in Section 3.2 in $q = p/2$ passes over a row-order stream, i.e. a sligthly weaker version of Theorem 1.1. As mentioned, in Section 4.3 we explain how to reduce the number of passes to $\lfloor p/4 \rfloor + 1$ using a small adjustment to the algorithm. Algorithm 1, computes multiple copies of the estimator in parallel using space $O(k)$ for each copy.

Algorithm 1 Algorithm for Schatten p-Norm of k-Sparse Matrices for $p \in 2\mathbb{Z}_{\geq 2}$ in Row-Order Streams

```
Input: $A \in \mathbb{R}^{n\times n}$ streamed in row-order, $p \in 2\mathbb{Z}_{\geq 2}$, $\varepsilon > 0$, $m \in \mathbb{Z}^+$.

in parallel $m$ times do                                                ▷ Each copy is a "walk"
    $i_1, \dots, i_q \leftarrow 0$, $Y_1, \dots, Y_q \leftarrow 0$
    in pass 1 do
        sample one row $i_1 \in [n]$ with probability $\frac{\|a_{i_1}\|_2^p}{\sum_j \|a_j\|_2^p}$        ▷ Using Reservoir Sampling
        $Y_1 \leftarrow \frac{\sum_j \|a_j\|_2^p}{\|a_i\|_2^p}$
    in pass $2 \leq t \leq q-1$ do
        sample one row $i_t \in [n]$ with probability $p^{i_1}_{i_{t-1}}(i)$        ▷ As defined in Section 3.2
        $Y_t \leftarrow Y_{t-1} \cdot \frac{\langle a_{i_{t-1}}, a_{i_t} \rangle}{p^{i_1}_{i_{t-1}}(i)}$
    in pass $q$ do
        compute $Y_q \leftarrow Y_{q-1} \sum_{i_q \in N_S^{i_1}(i_1)} \langle a_{i_{q-1}}, a_{i_q} \rangle \langle a_{i_q}, a_{i_1} \rangle c(i_1, \dots, i_q)$
return average of the $m$ copies of $Y_q$
```

Proof of Theorem 1.1 (version with $p/2$ passes). Algorithm 1 computes the estimator defined in Section 3.2 m times in parallel and outputs the average which we will denote by $\bar{Y}(A)$. Since the variance of the estimator is at most $C_p k^{\frac{3p}{2}-4}$ as per Theorem 3.5, by setting $m = \frac{Ck^{\frac{3p}{2}-4}}{\varepsilon^2}$ and the constant C appropriately, the guarantee on the estimate follows by an application of Chebyshev's Inequality to $\bar{Y}(A)$.

In pass t, each instance of the m parallel instances store the row a_{i_t} along with other estimates that can be stored in $O_p(1)$ words of space. Thus the total space complexity of the algorithm is $mk = O_p(\varepsilon^{-2}k^{\frac{3p}{2}-3})$ words. □

The proof of Theorem 4.1 (version with $p/2$ passes) follows the above by adjusting m according to Theorem 3.6.

4.2 Turnstile Streams

4.2.1 Preliminaries for Approximate Sampling

We define approximate samplers which we will use in turnstile streams to implement our estimator. Approximate L_p samplers have been studied extensively, see e.g. [CJ19].

Definition 4.3 (Approximate L_t Sampler)**.** Let $x \in \mathbb{R}^n$ be a vector and $t \geq 0$. An *approximate* L_t *sampler* with relative error ε, additive error Δ, and success probability $1-\delta$ is an algorithm that outputs each index $i \in [n]$ with probability

$$p_i \in (1 \pm \varepsilon)\frac{|x_i|^t}{\|x\|_t^t} \pm \Delta,$$

and with probability δ the sampler is allowed to output FAIL.

If an approximate sampler has no relative error and its additive error is less than n^{-C}, for arbitrarily large constant $C > 0$, then it is referred to as an *exact* L_p-sampler.

Generalizing L_p-samplers, we define approximate $L_{p,q}$-samplers for matrices.

Definition 4.4 (Weak Approximate $L_{t,q}$ Sampler)**.** Let $t, q \geq 0$ be constants and $A \in \mathbb{R}^{n \times m}$ be a matrix with rows $a_1, \ldots, a_n$. An *approximate* $L_{t,q}$ *sampler* with relative error ε, additive error Δ, and success probability $1-\delta$ is an algorithm that, conditioned on succeeding, outputs each index $i \in [n]$ with probability

$$p_i \in (1 \pm \varepsilon)\frac{\|a_i\|_q^t}{\sum_{j \in [n]} \|a_j\|_q^t} \pm \Delta,$$

and on failing, which occurs with probability δ, outputs any index.

We draw the attention of the reader to the success condition of the $L_{p,q}$ sampler; unlike for L_p samplers, the above definition is a weaker guarantee but is sufficient for our purpose since we can absorb the probability of failure for the sampler into the failure probability of the Schatten p-norm algorithm.

We recall some properties of higher powers of Gaussian distributions which we will use later in the analysis of sampling subroutines that we build. First, we give the higher moments of mean zero Gaussian random variables.

Fact 4.5. *For $t \geq 0$, $r \in 2\mathbb{Z}_{\geq 1}$ and a random variable $X \sim \mathcal{N}(0, t^2)$, we have*

$$\mathbb{E}\left[|X|^r\right] = t^r(r-1)!!.$$

We state a concentration property for polynomial functions of independent Gaussian/Rademacher random variables called Hypercontractivity Inequalities. For an introduction to the theory of hypercontractivity, see e.g. Chapter 9 of [O'D14].

Proposition 4.6 (Hypercontractivity Concentration Inequality, Theorem 1.9 [SS12])**.** *Consider a degree d polynomial $f(Y) = f(Y_1, \dots, Y_n)$ of independent centered Gaussian or Rademacher random variables $Y_1, \dots, Y_n$. Denote the variance $\sigma^2 = \mathrm{Var}\,(f(Y))$, then,*

$$\forall \lambda \geq 0, \quad \mathbb{P}\left[|f(Y) - \mathbb{E}\left[f(Y)\right]| \geq \lambda\right] \leq e^2 \exp\left(-\left(\frac{\lambda^2}{R \cdot \sigma^2}\right)^{\frac{1}{d}}\right)$$

where $R = R(d) > 0$ depends only on d.

4.2.2 Weak Sampler for Cascaded Norm $L_{p,2}$

Before giving our construction for approximate $L_{p,2}$ samplers in the turnstile model (Theorem 4.8), we recall some core results for L_p samplers that will be the algorithmic workhorse of our subroutine for $L_{p,2}$ sampling.

One can construct algorithms for approximate L_p samplers in various computational models. We look specifically at L_p samplers in the turnstile streaming model. The following algorithmic guarantees exist for approximate L_p samplers of vectors in turnstile streams.

Theorem 4.7 (Theorem 1.2 in [MW10])**.** *For $\delta > 0$ and $p \in 2\mathbb{Z}^+$, there exists an 0-relative-error L_p-sampler in turnstile streams, in 2-**passes**, with probability of outputting FAIL at most n^{-C} where $C > 0$ is an arbitrarily large constant. The algorithm uses $O_p(n^{1-2/p} \log^{O(p)} n)$ space.* [4]

For a given vector $x \in \mathbb{R}^n$ whose entries are streamed in a turnstile fashion, we will denote L_p-Sampler(x, δ) to be the output of the algorithm in Theorem 4.7 with failure probability at most δ. We will use this algorithm in turnstile streams for $p \geq 2$ to give an $O(\varepsilon)$ relative error $L_{p,2}$ sampler and failure probability at most δ for any given $\delta > 0$. The algorithm is fairly simple and is described in Algorithm 2.

Algorithm 2 Approximate $L_{p,2}$ Sampling Algorithm in Turnstile Streams

```
    INPUT: $A \in \mathbb{R}^{n \times n}$ as a turnstile stream, $p \in \mathbb{Z}_{\geq 2}$, $\hat{\delta} \in (0,1), \varepsilon > 0$.
1: Set $\hat{C}_p > 0$, $m \leftarrow \frac{\hat{C}_p \log^p n}{\varepsilon^2}$        ▷ $\hat{C}_p$ depends only on $p$
2: construct $G \in \mathbb{R}^{n \times m}$, with i.i.d standard Gaussian entries        ▷ drawn pseudorandomly
3: compute matrix $X \leftarrow \frac{1}{(p-1)!!} \cdot AG$
4: $(i,j) \leftarrow L_p$-Sampler$(x, \hat{\delta})$        ▷ where $x \in \mathbb{R}^{n^2}$ is the "flattened" version of $X$
5: return $i$ if $L_p$-sampler didn't output FAIL otherwise return any index
```

The matrix X, defined on line 3 in the above algorithm, can be computed "on the fly" given updates to A in the stream.

We then give the following theorem for approximate $L_{p,2}$ sampling in turnstile streams by arguing for the vector x defined in Algorithm 2, the average of the p^{th} power of the entries corresponding to row i is tightly concentrated around $\|a_i\|_2^p$.

Theorem 4.8. *For every $\varepsilon, C > 0, \delta \in (0,1)$ and $p \in 2\mathbb{Z}_{\geq 2}$, Algorithm 2 is an $O(\varepsilon)$ relative error and $O(n^{-C})$ additive error $L_{p,2}$ weak sampler in turnstile streams with failure probability at most δ. The algorithm uses $O_p(n^{1-2/p}\varepsilon^{-2}\log(\frac{1}{\delta})\log^{O(p)}(n))$ words of space.*

[4] The original theorem statement in the paper is for $p \in [0,2]$ but it is well-known among experts that the result extends to $p > 2$.

Proof. For a fixed $i \in [n]$, notice that $x_{i,1}, \ldots, x_{i,m}$ are independent and identically distributed as $\mathcal{N}\left(0, \frac{\|a_i\|_2^2}{((p-1)!!)^2}\right)$. Using Fact 4.5, $\mathbb{E}\left[x_{i,j}^p\right] = \|a_i\|_2^p$ for all $j \in [m]$ since p is even.

Let i^* be the output of Algorithm 2. From the guarantee for L_p-samplers by Theorem 4.7, conditioning on the L_p sampler succeeding, and setting the additive error sufficiently low, the probability that $i^* = i$ is

$$\mathbb{P}\left[i^* = i\right] = \sum_{j=1}^{m} \frac{x_{i,j}^p}{\|x\|_p^p} \pm O\left(n^{-C}\right).$$

We will first show that, for a fixed $i \in [n]$, the quantity $\sum_{j=1}^{m} x_{i,j}^p$ is tightly concentrated around $m\|a_i\|_2^p$ with high probability *over the randomness of the Gaussian sketch.*

Set the polynomial $f : \mathbb{R}^m \to \mathbb{R}$ on the random variables $\{x_{i,j}\}_{j=1}^m$ to be $f(x_{i,1}, \ldots, x_{i,m}) = \sum_{j=1}^{m} x_{i,j}^m$. Since the random variables $\{x_{i,j}\}_{j=1}^m$ are independent,

$$\mathrm{Var}(f(x_{i,1}, \ldots, x_{i,m})) = m\,\mathrm{Var}(x_{i,*}^p) = m\|a_i\|_2^{2p} \frac{(2p-1)!! - ((p-1)!!)^2}{((p-1)!!)^2}$$

for even $p > 2$. Using this to apply the Hypercontractivity Concentration Inequality for Gaussian random variables given in Proposition 4.6 gives us,

$$\mathbb{P}\left[\left|\sum_{j=1}^{m} x_{i,j}^p - m\|a_i\|_2^p\right| \geq \varepsilon m\|a_i\|_2^p\right] \leq e^2 \exp\left(-\left(\frac{\varepsilon^2 m}{C_p}\right)^{\frac{1}{p}}\right)$$

where C_p is a constant only dependent on p.

By setting $\hat{C}_p$ in Algorithm 2 appropriately, we can apply the the union bound over all $i \in [n]$ to obtain,

$$\mathbb{P}\left[i^* = i\right] = \frac{(1 \pm O(\varepsilon))\|a_i\|_2^p}{(1 \pm O(\varepsilon))\sum_{l=1}^{n} \|a_l\|_2^p} \pm O(n^{-C}) \qquad \text{for all } i \in [n]$$

with probability at least $1 - \hat{\delta} - n^{-\hat{c}}$ (where $\hat{c}$ is dependent on $\hat{C}_p$). Setting $\hat{\delta}$ appropriately in Algorithm 2 gives us the theorem. □

4.2.3 Recovering Rows and Their Neighbors

We also give some subroutines to recover rows and their neighbors so that we can compute inner-products between rows, sample neighbors and compute the probabilities for the estimator. The algorithmic core for these subroutines will be sparse-recovery algorithms which can be implemented using the Count-Sketch data structure described below.

Theorem 4.9 (Count-Sketch [CCF04])**.** *For all $w, n \in \mathbb{Z}^+$ and $\delta \in (0, 1)$, there is a randomized linear function $M : \mathbb{R}^n \leftarrow \mathbb{R}^s$ with $S = O(w \log(n/\delta))$ and a recovery algorithm A satisfying the following. For input $x \in \mathbb{R}^n$, algorithm A reads Mx and outputs a vector $\tilde{x} \in \mathbb{R}^n$ such that*

$$\forall x \in \mathbb{R}^n, \quad \mathbb{P}\left[\|x - \tilde{x}\|_\infty \leq \frac{1}{\sqrt{w}} \min_{x' : \|x'\|_0 = w} \|x - x'\|_2\right] \geq 1 - \delta.$$

Denote the output of a Count-Sketch algorithm on vector $x \in \mathbb{R}^n$ with parameter $w \in \mathbb{Z}^+$ and failure probability $\delta \geq 0$ to be $\textsc{Count-Sketch}_w(x, \delta)$. Notice that if it is guaranteed that x is k-sparse, i.e. $\|x\|_0 \leq k$, then the output $\textsc{Count-Sketch}_k(x, \delta)$ recovers the vector x exactly with probability at least $1 - \delta$ because $\min_{\tilde{x} : \|\tilde{x}\|_0 = k} \|x - \tilde{x}\|_2 = 0$ for every k-sparse vector x.

Reverting to our setting of k-sparse matrices in turnstile streams, given a target index $i \in [n]$, it is clear how to recover row a_i using $\tilde{O}(k)$ space using the Count-Sketch algorithm stated. Given a row a_i, we can recover the neighboring rows $\{a_j : j \in N(i)\}$ by running $\textsc{Count-Sketch}_k(A_{*,j}, \tilde{\delta})$ for each $j \in \text{supp}(a_i)$ (where $A_{*,j}$ corresponds to the j^{th} column of A). Since each column and row is k-sparse, with $\tilde{O}(k^2)$ space, we can recover the neighbors of row a_i given access to a_i. In addition, by setting the failure probability to $\frac{\delta}{k+1}$ in the above calls to $\textsc{Count-Sketch}_k$, our recovery subroutine will succeed with probability at least $1 - \delta$.

4.2.4 Algorithm for Turnstile Streams

We are now ready to present the algorithm implementing the estimator stated in Section 3.2 for turnstile streams. We note that unlike in row-order streams, we cannot recover the probability of sampling the first row exactly in turnstile streams. Since the output probability of the samplers is approximate, it introduces some bias in the estimator which we will have to bound. Therefore, the proof of correctness for this algorithm slightly deviates from that given in Theorem 2.2 but uses the same underlying ideas.

Let us introduce notation for the subroutines we will need. Denote by $L_{p,2}$-$\textsc{Sampler}(A, \varepsilon, \delta)$ the output of the approximate $L_{p,2}$ sampler defined in Algorithm 2 with relative error ε, and failure probability δ. Additionally, we will need to estimate the cascaded norm $L_{p,2}$ of A in order to bias the quantity we sample in our importance sampling estimator. Denote by $L_{p,2}$-$\textsc{NormEstimator}(A, \varepsilon, \delta)$ the output of an algorithm for estimating the $L_{p,2}$-norm of A with relative error ε and failure probability δ, such as in Section 4 of [JW09].

We describe our algorithm for turnstile streams in Algorithm 3, which runs $p + 1$ passes over the data, i.e. a sligthly weaker version of Theorem 4.2. As mentioned, the number of passes can be reduced to $\lfloor p/2 \rfloor + 3$ using the extra insight of Section 4.3.

Algorithm 3 Algorithm for Schatten p-norm of k-Sparse Matrices for $p \in 2\mathbb{Z}_{\geq 2}$ in Turnstile Streams

```
    Input: $A \in \mathbb{R}^{n \times n}$ in a stream with turnstile updates, $p \in 2\mathbb{Z}_{\geq 2}$, $\varepsilon > 0$, $m \in \mathbb{Z}^+$.

 1: in parallel $m$ times do
 2:     $i_1, \dots, i_q \leftarrow 0$, $Y_1, \dots, Y_q \leftarrow 0$
 3:     in stage 1 do                                                    ▷ takes 3 passes
 4:         $i_1 \leftarrow L_{p,2}$-SAMPLER$(A, \frac{\varepsilon}{k^{3p/4-2}}, \frac{1}{100})$
 5:         $\tilde{a}_{i_1} \leftarrow$ COUNT-SKETCH$_k(a_{i_1}, \frac{1}{100})$
 6:         $D_1 \leftarrow L_{p,2}$-NORMESTIMATOR$(A, \varepsilon, \frac{1}{100})$
 7:         $Y_1 \leftarrow \frac{D_1}{\|\tilde{a}_{i_1}\|_2^p}$
 8:     in stage $2 \leq t \leq q-1$ do                                   ▷ each stage takes 2 passes
 9:         $\tilde{C}_{t-1} \leftarrow \{$COUNT-SKETCH$_k(A_{*,j}, \frac{1}{100kq}) : j \in \text{supp}(\tilde{a}_{i_{t-1}})\}$
10:         reconstruct rows $\tilde{R}_{t-1} \leftarrow \{r_j : \text{row } j \text{ has support in } \tilde{C}_{t-1} \text{ and has } l_2\text{-norm less than } \tilde{a}_{i_1}\}$.
11:         $D_t \leftarrow \sum_{j \in \tilde{R}_{t-1}} \langle \tilde{a}_{i_{t-1}}, r_j \rangle$
12:         sample row index $i_t \in \text{supp}(\tilde{R}_{t-1})$ with probability $\frac{\langle \tilde{a}_{i_{t-1}}, r_{i_t} \rangle}{D_t}$
13:         $\tilde{a}_{i_t} \leftarrow$ COUNT-SKETCH$_k(a_{i_t}, \frac{1}{100q})$
14:         $Y_t \leftarrow Y_{t-1} \cdot \frac{D_t}{\langle \tilde{a}_{i_{t-1}}, \tilde{a}_{i_t} \rangle} \cdot \langle \tilde{a}_{i_{t-1}}, \tilde{a}_{i_t} \rangle$
15:     in stage $q$ do
16:         $\tilde{C}_{q-1} \leftarrow \{$COUNT-SKETCH$_k(A_{*,j}, \frac{1}{100k}) : j \in \text{supp}(\tilde{a}_{i_{q-1}})\}$
17:         reconstruct rows $\tilde{R}_{q-1} \leftarrow \{r_j : \text{row } j \text{ has support in } \tilde{C}_{q-1} \text{ and has } l_2\text{-norm less than } \tilde{a}_{i_1}\}$.
18:         compute
                $$Y_q \leftarrow Y_{q-1} \sum_{r_j \in \tilde{R}_{q-1}} \langle \tilde{a}_{i_{q-1}}, r_j \rangle \langle r_j, \tilde{a}_{i_1} \rangle c(i_1, \dots, i_{q-1}, j)$$
19: return average of the $m$ copies of $Y_q$
```

Proof of Theorem 4.2 (version with $p+1$ passes). Recall from Section 4.2.3 that COUNT-SKETCH$_k$ will recover all the entries of a k-sparse vector exactly with high probability. By setting the failure probability of each call to COUNT-SKETCH$_k$ to be sufficiently low, we can apply a union bound over all executions and assume that the algorithm recovers all the rows denoted by $\tilde{a}$ and r.

Let us assume that the L_p-sampler and Count-Sketch routines succeed and argue that taking the expectation over the randomness of the Gaussian sketch in the $L_{p,2}$-Sampler algorithm, the $L_{p,2}$-NORMESTIMATOR and the importance sampling estimator gives us that $|\mathbb{E}\left[\bar{Y}(A)\right] - \|A\|_{S_p}^p| \leq O_p(\varepsilon)\|A\|_{S_p}^p$.

Recall that the algorithm invokes an $O\left(\frac{\varepsilon}{k^{3p/4-2}}\right)$ relative error $L_{p,2}$-sampler in line 4. Since the additive error is less than n^{-C} for arbitrary $C \geq 0$, we can simply absorb it in the failure probability of the algorithm. We thus get,

$$\mathbb{E}\left[\bar{Y}(A)\right] = \sum_{\substack{(i_1,\dots,i_{q-1}) \\ \in \mathcal{S}}} \left(1 \pm \frac{O(\varepsilon)}{k^{3p/4-2}}\right) \frac{\|a_{i_1}\|_2^p}{\sum_j \|a_j\|_2^p} \frac{\mathbb{E}\left[D_1\right]}{\|a_{i_1}\|_2^p} \sum_{i_q \in N_S^{i_1}(i_1)} \sigma(i_1, \dots, i_q, i_1) c(i_1, \dots, i_q)$$

Since $L_{p,2}$-NormEstimator is an unbiased estimator for the $L_{p,2}$-norm, i.e. $\mathbb{E}[D_1] = \sum_j \|a_j\|_2^p$, we get

$$\left|\mathbb{E}\left[\bar{Y}(A)\right] - \|A\|_{S_p}^p\right| \leq \sum_{\substack{(i_1,\ldots,i_{q-1})\\ \in\mathcal{S}}} \frac{O(\varepsilon)}{k^{3p/4-2}} \left| \sum_{i_q \in N_S^{i_1}(i_1)} \sigma(i_1,\ldots,i_q,i_1)c(i_1,\ldots,i_q)\right|$$

We can upper bound the second term as we did in bounding the variance of the estimator in Theorem 2.2 to get $\left|\mathbb{E}\left[\bar{Y}(A)\right] - \|A\|_{S_p}^p\right| \leq O_p(\varepsilon)\|A\|_{S_p}^p$

It is left to bound the variance of $\bar{Y}(A)$. Again, we assume that the L_p-Sampler and Count-Sketch routines succeed and recall that that for a sequence $(i_1,\ldots,i_{q-1}) \in \mathcal{S}$, we define $z_{(i_1,\ldots,i_{q-1})} = \sum_{i_q \in N_S^{i_1}(i_1)} \sigma(i_1,\ldots,i_q,i_1)c(i_1,\ldots,i_q)$. Given the guarantee of $L_{p,2}$ sampling in Theorem 4.8, the variance of the estimate $\bar{Y}(A)$ is

$$\mathrm{Var}\left(\bar{Y}(A)\right) \leq \frac{1}{m} \sum_{\substack{(i_1,\ldots,i_{q-1})\\ \in\mathcal{S}}} (1 \pm \frac{O(\varepsilon)}{k^{3p/4-2}}) \frac{1}{\sum_j \|a_j\|_2^p} \frac{\mathbb{E}\left[D_1^2\right]}{\|a_{i_1}\|_2^p} \prod_{t=2}^{q-1} \frac{1}{p_{i_{t-1}}^{i_1}(i_t)} \left(z_{(i_1,\ldots,i_{q-1})}\right)^2$$

By the accuracy guarantee of $L_{p,2}$-NormEstimator and Fact 2.1,

$$\leq \frac{1}{m} \sum_{\substack{(i_1,\ldots,i_{q-1})\\ \in\mathcal{S}}} (1 \pm O(\varepsilon)) \frac{\|A\|_{S_p}^p}{\|a_{i_1}\|_2^p} \prod_{t=2}^{q-1} \frac{1}{p_{i_{t-1}}^{i_1}(i_t)} \left(z_{(i_1,\ldots,i_{q-1})}\right)^2$$

Bounding this identically as we did in Theorem 2.2 and setting $m = \frac{Ck^{3p/2-4}}{\varepsilon^2}$ give us $\mathrm{Var}(\bar{Y}(A)) \leq C_p\varepsilon\|A\|_{S_p}^{2p}$ where C_p is a constant dependent only on p.

The $L_{p,2}$-Sampler with $O\left(\frac{\varepsilon}{k^{3p/4-2}}\right)$ relative error takes space $\tilde{O}_p(k^{\frac{3p}{2}-4} n^{1-\frac{2}{p}} (\varepsilon^{-1}\log n)^{O(p)})$ and the $L_{p,2}$-NormEstimator takes space $\tilde{O}_p(n^{1-\frac{2}{p}}(\varepsilon^{-1}\log n)^{O(p)})$. In addition, storing the rows recovered from Count-Sketch requires $\tilde{O}(k^2)$ space. Thus, the space complexity of repeating the estimator $m = \frac{Ck^{3p/2-4}}{\varepsilon^2}$ times is $\tilde{O}_p(k^{3p-6} n^{1-\frac{2}{p}}(\varepsilon^{-1}\log n)^{O(p)})$. We note that in stage 1, the sampler takes two passes, followed by another pass for Count-Sketch and the norm estimator. The subsequent stages requires two passes each giving a total of $3 + 2(q-1) = p+1$ passes. □

4.3 Fewer Passes

As mentioned earlier, we can slightly modify the way we implement the estimator from Section 3.2 to reduce the number of passes that Algorithm 1 and Algorithm 3 make to $\lfloor\frac{p}{4}\rfloor + 1$ and $\frac{p}{2} + 3$, respectively. This is explained below and completes the proofs of Theorems 1.1, 4.1 and 4.2.

The idea is to sample each sequence $(i_1,\ldots,i_q) \in \mathcal{S}$ in a different way albeit with the same probability. Assume for simplicity that $p \equiv 0 \pmod 4$. After sampling the first row $i_1 \in [n]$, we sample independently two "paths" of length $p/4-1$, each starting at i_1, with probabilities identical to the ones in the estimator. We then sum over the common neighbors of the endpoints of the two paths, using each of them to complete a cycle of length $p/2$. Formally, sample independently two sequences of rows $(i_1,l_1,\ldots,l_{q/2-1}),(i_1,j_1,\ldots,j_{q/2-1}) \in \Gamma_S^{i_1}(i_1,q/2-1)$. Denote by r the sequence of rows $(l_{q/2-1},\ldots,l_1,i_1,j_1,\ldots,j_{q/2-1})$ then the following estimator is equivalent to the estimator described in Section 3.2 (slightly abusing the notation therein for concatenating two sequences of rows):

$$Y := \frac{1}{\tau_r} \sum_{\substack{m \in N_S^{i_1}(l_{q/2-1}) \\ \cap N_S^{i_1}(j_{q/2-1})}} c(r, i_q)\sigma(r)\langle a_{l_{q/2-1}}, a_m\rangle\langle a_{j_{q/2-1}}, a_m\rangle.$$

It is easy to verify that this estimator is unbiased, and that its variance can be bounded using the proof steps of Section 3.2. This estimator can be implemented algorithmically similarly to the description in Sections 4.1 and 4.2 using less passes over the stream. Specifically, the above approach decreases the number of "path" stages (i.e. all but the "seed" sampling stage) by a factor of (roughly) 2, and the space complexity remains the same up to constant factors. Therefore, we reduce the number of passes over the streams of Algorithm 1 and Algorithm 3 to $\lfloor \frac{p}{4} \rfloor + 1$ and $\frac{p}{2} + 3$, respectively. This concludes the proofs of Theorems 1.1, 4.1 and 4.2.

5 Pass-Space Trade-off

Very often streaming problems have a sharp transition in space complexity when comparing a single pass to multiple passes over the data. However, it turns out that for the Schatten p-norm of sparse matrices, the space dependence on the number of passes is smooth, allowing one to pick the desired pass-space trade-off. Specifically, for any parameter $s \geq 2$, one can $(1 \pm \varepsilon)$-approximate the Schatten p-norm in $\lfloor \frac{p}{2(s+1)} \rfloor + 1$ passes using $O_{p,s}(\varepsilon^{-3}k^{2ps}n^{1-\frac{1}{s}})$ words of space.

Recall the Schatten p-norm formulation of (3.3). This in can be interpreted as partitioning the (contributing) length-q cycles according to their heaviest row, denoted here by i_1. Analogously, for any parameter $s \in [2, p-1]$, we split the cycle into $s+1$ segments of hop-distance roughly $\frac{q}{s+1}$, and further partition the cycles according to the heaviest row in each such segment. The idea is to "cover" a cycle by sampling s rows, where each sampled row is the heaviest among its segment. More precisely, each sample "covers" its segment, except for the heaviest row in the entire cycle that will "cover" two segments. Then, to evaluate the entire cycle we need $\lfloor \frac{q}{s+1} \rfloor + 1$ passes. The total space needed by the algorithm is $O_{p,s}(\varepsilon^{-3}k^{2ps}n^{1-1/s})$ words of space, mostly as it computes multiple copies of the estimator (to reduce the variance), similarly to Section 4.

In the first subsection we focus on the case $s = 2$ and present a BFS-based algorithm, followed by a brief explanation how to improve the dependence on k by replacing the BFS with adaptive sampling as in the previous sections. In the second subsection we generalize the result to any $s \geq 2$.

5.1 The Basic Case $s = 2$ ($\lfloor \frac{p}{6} \rfloor + 1$ Passes)

As mentioned, (3.3) can be interpreted as considering only cycles that "start" from the heaviest row of the cycle (by "rotating" the cycle). We suggest a variation on this idea. Given a q-cycle "starting" at the heaviest row i, we identify the row j that is the heaviest among the rows at least $q/3$ cycle-hops away from i. In other words, if the cycle is $(i = i_1, \ldots, i_q)$, then j is the heaviest among (roughly) $i_{q/3}, \ldots, i_{2q/3}$. Therefore, our aim is to sample rows i and j and then to connect four paths: two starting from i and two starting from j, each of hop-distance at most $q/3$. As we don't know in advance the hop-distance to row j, we store all possible options and only later decide which paths to stich together into a cycle.

Formally, we augment the notation of paths presented in Section 3. For indices $i, j, i_1 \in [n]$ and integers $t' \leq t'' \leq t$, define

$$\Gamma_S^{(i,j;t',t'')}(i_1, t) :=$$

$$\Big\{(i_1,\dots,i_q):\ (i_1,\dots,i_{t'})\in\Gamma_S^i(i_1,t'),(i_{t'},\dots,i_{t''})\in\Gamma_S^j(i_{t'},t''-t'+1),(i_{t''},\dots,i_t)\in\Gamma_S^i(i''_t,t-t''+1)\Big\}.$$

As we are actually interested in the special case where $t'=\lfloor\frac{q}{3}\rfloor+1$ and $t''=q-\lfloor\frac{q}{3}\rfloor$, we shall omit t',t'' from the superscript in this special case.

Recall that we focus on cycles in which $i_1=i$, i.e. the heaviest row is the starting of the cycle. Furthermore, we want $j=i_l$ for some $l\in\{\lfloor\frac{q}{3}\rfloor+2,\dots,q-\lfloor\frac{q}{3}\rfloor\}$, i.e. j is part of the cycle, and is at least $\lfloor\frac{q}{3}\rfloor$ cycle-hops away from i. Accordingly, we can rewrite the Schatten p-norm as

$$\|A\|_{S_p}^p=\sum_{i,j}\ \sum_{\lfloor\frac{q}{3}\rfloor+2\le l\le q-\lfloor\frac{q}{3}\rfloor}\ \sum_{\substack{(i,i_2,\dots,i_q)\\ \in\Gamma_S^{(i,j)}(i):\ i_l=j}} c(i,i_2,\dots,i_q)\sigma(i,i_2,\dots,i_q,i). \tag{5.1}$$

We are now ready to present our estimator and an algorithm implementing it. In the algorithm, instead of summing over all $i,j\in[n]$, we sample two multisets I,J and do a BFS of depth $\lfloor q/3\rfloor$ from each $i\in I$ and $j\in J$, and eventually enumerate over all cycles involving these i,j as in (5.1).

Algorithm 4 Two-Set based Algorithm for Schatten p-Norm of k-Sparse Matrices for $p\in2\mathbb{Z}_{\ge2}$ in Row-Order Stream

```
Input: $A\in\mathbb{R}^{n\times n}$ streamed in row-order, $p\in2\mathbb{Z}_{\ge2}$, $\varepsilon>0$.

1: $r\leftarrow O(\varepsilon^{-3}q^{5/2}k^{3p-6}\sqrt{n})$, $Y\leftarrow 0$.
2: in parallel $2r$ times do
3:     in pass 1 do
4:         sample a row $i\in[n]$ with probability $\tau_i=\frac{\|a_i\|_2^q}{\sum_m\|a_m\|_2^q}$    ▷ Using Reservoir Sampling
5:     in pass $2\le t\le\lfloor q/3\rfloor+1$ do
6:         store all rows of hop-distance at most $t-1$ from $i$ that have $l_2$-norm smaller than row $i$
7: let multisets $I$ and $J$ contain the first and last $r$ samples (from line 4), respectively
8: for each $(i,j)\in I\times J$ such that $\left(\frac{\varepsilon}{qk^{2\lceil q/2\rceil}}\right)^{3/p}\|a_i\|_2\le\|a_j\|_2\le\|a_i\|_2$ do
       $$Y \mathrel{+}= \frac{1}{\tau_i\cdot\tau_j}\sum_{\lfloor\frac{q}{3}\rfloor+2\le l\le q-\lfloor\frac{q}{3}\rfloor}\ \sum_{\substack{(i,i_2,\dots,i_q)\\ \in\Gamma_S^{(i,j)}(i):\ i_l=j}} c(i,i_2,\dots,i_q)\sigma(i,i_2,\dots,i_q,i)$$
9: return $\bar{Y}=\frac{1}{r^2}Y$
```

Theorem 5.1. *There exists an algorithm that, given $p\in2\mathbb{Z}_{\ge2}$, $\varepsilon>0$ and a k-sparse matrix $A\in\mathbb{R}^{n\times n}$ that is streamed in row-order, makes $\lfloor\frac{p}{6}\rfloor+1$ passes over the stream using at most $O_p(\varepsilon^{-3}k^{4p}\sqrt{n})$ words of space, and then outputs $\bar{Y}(A)$ that $(1\pm2\varepsilon)$-approximates $\|A\|_{S_p}^p$ with probability at least $2/3$.*

Before the proof, we state the following theorem, which can be viewed as a variant of the Importance Sampling lemma (Theorem 2.2).

Lemma 5.2 (Two-Set Sampling)**.** *Let $z=\sum_{i,j\in[n]}z_{i,j}>0$ for $n\ge1$, and suppose the matrix defined by $\{z_{i,j}\}$ is Δ-sparse.*[5] *Let $I,J\in[n]$ be two random multisets of size r, where each of their*

[5] Δ can be viewed as an upper bound on the in-degrees and out-degrees of the directed graph defined by edge weights z_{ij} on vertex set $[n]$.

$2r$ elements is chosen independently with replacement according to the distribution $(\tau_l : l \in [n])$. Consider the estimator

$$Y = \frac{1}{r^2} \sum_{i \in I, j \in J} \frac{z_{i,j}}{\tau_i \cdot \tau_j}.$$

If $\lambda > 0$ satisfies that for all $i, j \in [n]$ both $\tau_i, \tau_j \geq \frac{1}{\lambda}\sqrt{\frac{|z_{i,j}|}{z}}$, then

$$\mathbb{E}\left[Y\right] = z \quad and \quad \operatorname{Var}(Y) \leq \left(\frac{\lambda^2}{r^2} + \frac{2\lambda\Delta}{r}\right) z \sum_{i,j \in [n]} |z_{i,j}|. \tag{5.2}$$

The proof of Lemma 5.2 is given in Appendix A.3. We now proceed to the proof of Theorem 5.1, remarking that $k^{O(p)}$ factor can be improved by using the Projection Lemmas, but for simplicity we use more straightforward arguments.

Proof of Theorem 5.1. First we remark that indeed in $\lfloor q/3 \rfloor + 1$ passes all the needed rows of a cycle are kept. For any cycle, row i needs to “cover” $\lfloor q/3 \rfloor + 1 + (q - (q - \lfloor q/3 \rfloor)) = 2\lfloor q/3 \rfloor + 1$ rows (including itself), which indeed happens as we do a BFS of size $\lfloor q/3 \rfloor$. Row j must cover at most $q - \lfloor q/3 \rfloor - (\lfloor q/3 \rfloor + 2) = q - 2\lfloor q/3 \rfloor - 2$ rows, including itself. As $\lfloor q/3 \rfloor + 1 \geq q - 2\lfloor q/3 \rfloor - 2$, we indeed again cover all possibly needed rows in the $\lfloor q/3 \rfloor + 1$ passes. We now go on to prove the approximation bounds. Let $\beta \coloneqq \left(\frac{\varepsilon}{qk^{p-2}}\right)^{3/p}$ and define for all $i, j \in [n]$

$$z_{i,j} \coloneqq \begin{cases} \sum_{\lfloor \frac{q}{3} \rfloor + 2 \leq l \leq q - \lfloor \frac{q}{3} \rfloor} \sum_{\substack{(i,i_2,\ldots,i_q) \\ \in \Gamma_S^{(i,j)}(i):\ i_l = j}} c(i, i_2, \ldots, i_q)\sigma(i, i_2, \ldots, i_q, i) & \text{if } \|a_j\|_2 \leq \|a_i\|_2; \\ 0 & \text{otherwise.} \end{cases}$$

Then, by Equation (5.1), $z' \coloneqq \sum_{i,j} z_{i,j} = \|A\|_{S_p}^p$. Since line 8 in the algorithm considers only pairs (i, j) where $\frac{\|a_j\|_2}{\|a_i\|_2} \in [\beta, 1]$, we further define

$$z \coloneqq \sum_{i,j:\ \frac{\|a_j\|_2}{\|a_i\|_2} \in [\beta, 1]} z_{i,j}.$$

Let us show that the omitted terms do not contribute much to $z' = \|A\|_{S_p}^p$, and thus the error introduced by omitting them is small. For simplicity assume $q/3 \in \mathbb{N}$, then

$$\begin{aligned} \left|z' - z\right| &\leq \sum_i \sum_{j:\ \frac{\|a_j\|_2}{\|a_i\|_2} \leq \beta} |z_{i,j}| \\ &\leq \sum_i \sum_{j:\ \frac{\|a_j\|_2}{\|a_i\|_2} \leq \beta} \sum_{\lfloor \frac{q}{3} \rfloor + 2 \leq l \leq q - \lfloor \frac{q}{3} \rfloor} \sum_{\substack{(i,i_2,\ldots,i_q) \\ \in \Gamma_S^{(i,j)}(i):\ i_l = j}} c(i, i_2, \ldots, i_q) \left|\sigma(i, i_2, \ldots, i_q, i)\right| \end{aligned}$$

As $c(i, i_2, \ldots, i_q) \leq q$, and using the conditions on i and j we get

$$\leq q \sum_i \sum_{j:\ \frac{\|a_j\|_2}{\|a_i\|_2} \leq \beta} \sum_{\lfloor \frac{q}{3} \rfloor + 2 \leq l \leq q - \lfloor \frac{q}{3} \rfloor} \sum_{\substack{(i,i_2,\ldots,i_q) \\ \in \Gamma_S^{(i,j)}(i):\ i_l = j}} \|a_i\|_2^{2p/3} \|a_j\|_2^{p/3}$$

As each row has at most k^2 "neighboring" rows,

$$\leq k^{2(q-1)} q \beta^{p/3} \sum_i \|a_i\|_2^p = \varepsilon \sum_i \|a_i\|_2^p.$$

Therefore, using Fact 2.1, we conclude

$$|z - \|A\|_{S_p}^p| \leq \varepsilon \|A\|_{S_p}^p. \tag{5.3}$$

We proceed to show that the standard deviation of our estimator is bounded by εz, meaning that w.h.p $\bar{Y} \in (1 \pm \varepsilon) z$, and together with (5.3) this yields $\bar{Y} \in (1 \pm 2\varepsilon)\|A\|_{S_p}^p$. To this end, we want to use Lemma 5.2 and thus wish to show that

$$\sum_{i,j} |z_{i,j}| \leq 2qk^{2\lceil q/2 \rceil} z \tag{5.4}$$

and that $\lambda \coloneqq \sqrt{2qk^{p-4}\frac{n}{\beta^{2p/3}}} = \sqrt{2} q^{3/2} k^{3p/2-4} \frac{\sqrt{n}}{\varepsilon}$ satisfies

$$\frac{|z_{i,j}|}{z} \leq \lambda^2 \tau_j^2 \qquad \forall i, j \in [n]. \tag{5.5}$$

meaning that . We remark that (5.5) is indeed sufficient, as $\tau_j \leq \tau_i$, as otherwise $z_{i,j} = 0$ and the inequality trivially holds.

To prove (5.4), we use similar arguments as above, together with (5.3),

$$\begin{aligned}
\sum_{i,j} |z_{i,j}| &\leq q \cdot \sum_{i,j:\ \frac{\|a_j\|_2}{\|a_i\|_2} \in [\beta,1]} \ \sum_{\lfloor \frac{q}{3} \rfloor + 2 \leq l \leq q - \lfloor \frac{q}{3} \rfloor} \ \sum_{\substack{(i,i_2,\dots,i_q) \\ \in \Gamma_S^{(i,j)}(i):\ i_l = j}} \|a_i\|_2^p \\
&\leq qk^{p-2} \sum_i \|a_i\|_2^p \\
&\leq 2qk^{p-2} z.
\end{aligned}$$

To prove (5.5), fix i, j such that $\frac{\|a_j\|}{\|a_i\|} \in [\beta, 1]$, then by similar arguments, together with (5.3) and Fact 2.1,

$$\begin{aligned}
\frac{|z_{i,j}|}{z} &\leq \frac{1}{z} \sum_{\lfloor \frac{q}{3} \rfloor + 2 \leq l \leq q - \lfloor \frac{q}{3} \rfloor} \ \sum_{\substack{(i,i_2,\dots,i_q) \\ \in \Gamma_S^{(i,j)}(i):\ i_l = j}} c(i, i_2, \dots, i_q) |\sigma(i, i_2, \dots, i_q, i)| \\
&\leq \frac{1}{z} \sum_{\lfloor \frac{q}{3} \rfloor + 2 \leq l \leq q - \lfloor \frac{q}{3} \rfloor} \ \sum_{\substack{(i,i_2,\dots,i_q) \\ \in \Gamma_S^{(i,j)}(i):\ i_l = j}} q \|a_i\|_2^{2p/3} \|a_j\|_2^{p/3} \\
&\leq qk^{p-4} \frac{\|a_j\|_2^p}{\beta^{2p/3} z} \\
&\leq 2qk^{p-4} \frac{\|a_j\|_2^p}{\beta^{2p/3} \|A\|_{S_p}^p} \\
&\leq 2qk^{p-4} \frac{\|a_j\|_2^p}{\beta^{2p/3} \sum_m \|a_m\|_2^p}
\end{aligned}$$

using norm properties (basically applying $\|v\|_q \le n^{1/q-1/p}\|v\|_p$ to the vector $v = (\|a_1\|_2, \dots, \|a_n\|_2)$),

$$\begin{aligned} &\le qk^{p-4}\frac{\|a_j\|_2^p}{\beta^{2p/3}(\sum_m \|a_m\|_2^q)^2/n} \\ &\le 2qk^{p-4}\frac{n}{\beta^{2p/3}} \cdot \tau_j^2. \end{aligned}$$

We further note that for $z_{i,j}$ to be non-zero, row j must be at distance at most $\lceil q/2 \rceil$ from row i, and thus each row can participate in at most $k^{2\lceil q/2 \rceil}$ different non-zero $z_{i,j}$, i.e., $\Delta \le k^{p/2-2}$. Combining all the above, we conclude that setting $r = O(\varepsilon^{-2}\lambda\Delta) \cdot 2qk^{p-2} = O(\varepsilon^{-3}q^{5/2}k^{3p-6}\sqrt{n})$ will give w.h.p a $(1 \pm 2\varepsilon)$-approximation to the Schatten p-norm by Chebyshev's inequality.

As for each row in $I \cup J$ the algorithm stores neighborhoods of size $O\left((k^2)^{q/3}\right)$, and storing each row in the neighborhood takes $O(k)$ words, there is an extra factor of $k^{p/3+1}$. Thus the total space is $O(\varepsilon^{-3}q^{5/2}k^{10p/3-5}\sqrt{n})$ words. □

Remark. As mentioned earlier, the BFS approach can be replaced with the adaptive sampling approach from previous sections. For the first r samples (in I), the algorithm adaptively samples two paths of hop-distance (roughly) $q/3$, similarly to Section 4.3. For each of the last r samples (in J), the algorithm chooses $\rho \in [q/3]$ uniformly at random (and independently of all other steps), and adaptively samples a path of hop-distance ρ and a path of hop-distance (roughly) $\frac{q}{3} - \rho$. It then tries to "stitch" these paths to create q-cycles. The bound on λ (i.e. (5.5)) increases by a factor of $q/3$ due to ρ (this can be viewed as replacing the BFS with multiple random paths), but as the algorithm does not keep the entire neighborhoods, a $k^{p/3}$ factor is shaved from the space complexity. This, together with a tighter analysis, can improve the dependence on k in Theorem 5.1 to $k^{19p/8+O(1)}$.

5.2 General s (using $\lfloor\frac{p}{2(s+1)}\rfloor + 1$ Passes)

We generalize the algorithm from the previous subsection, such that given some integer $s \in [2, p-1]$, the algorithm samples in parallel in the first pass $r \cdot s$ rows for $r = O_{p,\varepsilon,s}(k^{4p}n^{1-1/s})$, where each "seed" row i is sampled with probability $\tau_i = \frac{\|a_i\|_2^{p/s}}{\sum_m \|a_m\|_2^{p/s}}$. In the following passes it runs a BFS of depth (roughly) $\frac{q}{s+1}$, keeping all the shorter rows (in l_2-norm) in the neighborhood of each seed. The first r samples are denoted as multiset I, and the other samples are split into $s-1$ multisets of size r denoted as $J_1, \dots, J_{s-1}$. The algorithm then considers s-tuples $(i, j_1, \dots, j_{s-1})$ where $i \in I$ and every row $j_u \in J_u$ has l_2-norm in the range $(\beta', 1)$ relative to that of row i, for $\beta' \approx \left(\frac{\varepsilon}{sqk^p}\right)^{(s+1)/p}$. The estimator is formed by looking at the eligible s-tuples, and for each such tuple adding the contributions of all the q-cycles obtained by "stitching" paths of hop-distance (roughly) $\frac{q}{s+1}$ passing through these seeds, as follows:

$$Y \mathrel{+}= \frac{1}{\tau_i \tau_{j_1} \cdots \tau_{j_{s-1}}} \sum_{\frac{q}{s+1} \le l_1 \le \frac{2q}{s+1}} \cdots \sum_{\frac{(s-1)q}{s+1} \le l_{s-1} \le \frac{s \cdot q}{s+1}} \sum_{\substack{(i,i_2,\dots,i_q) \\ \in \Gamma_S^{(i,j_1,\dots,j_{s-1})}(i): \\ i_{l_1}=j_1,\dots,i_{l_{s-1}}=j_{s-1}}} c(i, i_2, \dots, i_q)\sigma(i, i_2, \dots, i_q, i).$$

The algorithm's final output is $\bar{Y} = \frac{1}{r^s}Y$.

Theorem 5.3. *There exists an algorithm that, given $p \in 2\mathbb{Z}_{\ge 2}$, $\varepsilon > 0$, an integer $s \in [2, p-1]$ and a k-sparse matrix $A \in \mathbb{R}^{n \times n}$ streamed in row-order, makes $\lfloor\frac{p}{2(s+1)}\rfloor + 1$ passes over the stream* *using* $O_p\left(\varepsilon^{-3}k^{2ps}n^{1-\frac{1}{s}}\right)$ *words of space, and outputs* $\bar{Y}(A)$ *that* $(1\pm 2\varepsilon)$*-approximates* $\|A\|_{S_p}^p$ *with probability at least* $2/3$*.*

Proof Sketch. The proof follows similar steps as the proof for $s=2$. First, the error introduced by taking only certain cycles changes, as now we miss cycles in which at least one of the sampled j_u is smaller than β'. However their total contribution can be bounded by $(s-1)(\beta')^{p/(s+1)}qk^{p-2}<\varepsilon$ relative to $\|A\|_{S_p}^p$. Next, an s-Set Sampling Lemma is proved using the same arguments as the Two-Set Sampling Lemma. It asserts that the estimator

$$Y=\frac{1}{r^s}\sum_{i\in I,j_1\in J_1,\dots,j_{s-1}\in J_{s-1}}\frac{z_{i,j_1,\dots,j_{s-1}}}{\tau_i\tau_{j_1}\cdots\tau_{j_{s-1}}}$$

is unbiased, and that if $\lambda>0$ satisfies that for every $i,j_1,\dots,j_{s-1}\in[n]$, all $\tau_i,\tau_{j_1},\dots,\tau_{j_{s-1}}\geq\frac{1}{\lambda}\left(\frac{|z_{i,j_1,\dots,j_{s-1}}|}{z}\right)^{1/s}$, then

$$\mathrm{Var}(Y)\leq O\left(\left(\Delta+\frac{\lambda}{r}\right)^s-\Delta^s\right)z\sum_{i,j_1,\dots,j_{s-1}\in[n]}|z_{i,j_1,\dots,j_{s-1}}|.$$

The proof for the inequality analogous to (5.4), which bounds the ratio between the absolute sum of $z_{i,j_1,\dots,j_{s-1}}$ and z, is the same. To prove the bound λ (i.e. analogous to (5.5)), we need to bound the shortest j_u among rows $(j_1,\dots,j_{s-1})$. To do so we first bound all "seeds" except j_u using row i, and then use the same arguments that result in $\lambda=\left(C_\varepsilon qk^p\frac{n^{s-1}}{(\beta')^{2p/(s+1)}}\right)^{1/s}$ for a suitable constant C dependent on ε. Finally, now each i can have $(s-1)k^{2\lceil q/2\rceil}$ different $(j_1,\dots,j_{s-1})$, i.e. $\Delta\leq(s-1)k^{q+2}$. Picking $r=O\left(\varepsilon^{-3}s\Delta^{s-1}\lambda\right)$ results in the desired approximation.

The space complexity analysis is as in the proof of Theorem 5.1, resulting in

$$O\left(\varepsilon^{-3}(s-1)^s\cdot q^{2+1/s}\cdot k^{p(s/2+11/6+1/s)+2s-O(1)}\cdot n^{1-1/s}\right)$$

words of space. □

6 Lower Bound for One-Pass Algorithms in the Row-Order Model

We show a space lower bound of $\Omega(n^{1-4/\lfloor p\rfloor_4})$ bits for one-pass algorithms and even p values in the row-order model. Our main technical contribution is the analysis of even p values in a reduction presented in [LW16a], based on the Boolean Hidden Hypermatching [VY11, BS15]. Although this is not mentioned in [LW16a], it can easily be verified from the proof of [LW16a, Theorem 3] (stated below as Theorem 6.1) that this reduction applies also to the row-order model.[6] Our bound is closely related to the $\Omega(n^{1-1/\varepsilon})$ bits lower bound for $p\notin 2\mathbb{Z}$, proved in [BCK+18], and is also nearly tight with the upper bound from the same paper (see discussion at the end of this section).

We first recall the definitions presented in [LW16a]. Let $D_{m,l}$ (for $0\leq l\leq m$) be an $m\times m$ diagonal matrix with the first l diagonal elements equal to 1 and the remaining diagonal entries equal to 0, and let $\mathbf{1}_m$ be an m-dimensional vector full of 1s, thus $\mathbf{1}_m\mathbf{1}_m^\top$ is the $m\times m$ all-ones matrix. Define

$$M_{m,l}(\gamma)=\begin{pmatrix}\mathbf{1}_m\mathbf{1}_m^\top & 0\\ \sqrt{\gamma}D_{m,l} & 0\end{pmatrix},$$

[6] In fact, also Theorem 4 in [LW16a] applies to row-order streams, providing a different proof for the $\Omega(n^{1-\varepsilon})$ lower bound for $p\notin 2\mathbb{Z}$ proved in [BCK+18].

where $\gamma > 0$ is a constant (which may depend on m).

Let $m \geq 2$ be an even integer, and let $p_m(l) := \binom{m}{l}/2^{m-1}$ for $0 \leq l \leq m$. Let $\mathcal{E}(m)$ be the probability distribution defined on even integers $\{0, 2, \ldots, m\}$ with probability density function $p_m(l)$. Similarly, let $\mathcal{O}(m)$ be the distribution on odd integers $\{1, 3, \ldots, m-1\}$ with density function $p_m(l)$. We say that a function f on square matrices is *diagonally block-additive* if $f(X) = f(X_1) + \ldots + f(X_s)$ for any block diagonal matrix X with square blocks $X_1, \ldots, X_s$. As noted in [LW16a], $f(X) = \|X\|_{S_p}^p$ is diagonally block-additive.

We observe that the reduction presented in [LW16a] is applicable also to row-order streams, and thus state below a slightly stronger version of Theorem 3 from that paper.

Theorem 6.1 (Theorem 3 in [LW16a])**.** *Let t be an even integer and let f be a function of square matrices that is diagonally block-additive. If there exists $m = m(t)$ and $\gamma = \gamma(m) > 0$, such that the following "gap condition" holds:*

$$\mathbb{E}_{l\sim\mathcal{E}(t)}\left[f\left(M_{m,l}(\gamma)\right)\right] - \mathbb{E}_{l\sim\mathcal{O}(t)}\left[f\left(M_{m,l}(\gamma)\right)\right] \neq 0, \tag{6.1}$$

*then there exists a constant $\varepsilon = \varepsilon(t) > 0$ such that every **row-order** streaming algorithm that, given $X \in \mathbb{R}^{N\times N}$ (for sufficiently large N), approximates $f(X)$ within factor $1 \pm \varepsilon$ with constant error probability, must use $\Omega_t(N^{1-1/t})$ bits of space.*

We can now present our analysis for even p values.

Lemma 6.2. *Let $f(X) = \|X\|_{S_p}^p$, for $p \in 4\mathbb{Z}_{\geq 1}$. Then the gap condition (6.1) is satisfied, under the choice $m = t$ and $\gamma = 1$, if and only if $t \leq p/4$.*

Proof. As shown in the proof of Theorem 4 in [LW16a], for $m = t$ and $\gamma = 1$, the non-zero singular values of a block $M_{t,l}(1)$ are as follows. For $l = 0$, the only non-zero singular value is t. For $0 < l < t$, the non-zero singular values are $r_1(l) = \sqrt{\frac{(t^2+1)+\sqrt{(t^2-1)^2+4lt}}{2}}$, $r_2(l) = \sqrt{\frac{(t^2+1)-\sqrt{(t^2-1)^2+4lt}}{2}}$ and 1 with multiplicity $l-1$. And for $l = t$, the non-zero singular values are $r_1(t) = \sqrt{\frac{(t^2+1)+\sqrt{(t^2-1)^2+4t^2}}{2}}$ and 1 with multiplicity $t-1$. Further note that that $r_2(t) = 0$. Using this, and recalling the distribution of the blocks, the left-hand side of the gap condition (6.1) is

$$\frac{1}{2^{t-1}}\left[t^p + \sum_{\text{even } l}\binom{t}{l}\left((l-1) + r_1^p(l) + r_2^p(l)\right) - \sum_{\text{odd } l}\binom{t}{l}\left((l-1) + r_1^p(l) + r_2^p(l)\right)\right] \tag{6.2}$$

and we can rewrite this as

$$\frac{1}{2^{t-1}}\left[t^p + \sum_{0<l\leq t}\binom{t}{l}(-1)^l(l-1) + \sum_{0<l\leq t}\binom{t}{l}(-1)^l\left(r_1^p(l) + r_2^p(l)\right)\right].$$

For the first sum, by Corollary 2 in [Rui96], we know that

$$\sum_{l=0}^{t}(-1)^l\binom{t}{l}(l-1) = 0$$

meaning that

$$\sum_{0<l\leq t}\binom{t}{l}(-1)^l(l-1) = 1.$$

Let $q = p/2$. It holds that

$$r_1^p(l)+r_2^p(l)=\left(\frac{(t^2+1)+\sqrt{(t^2-1)^2+4lt}}{2}\right)^q+\left(\frac{(t^2+1)-\sqrt{(t^2-1)^2+4lt}}{2}\right)^q$$

and using the binomial theorem,

$$=\frac{1}{2^q}\left[\sum_{i=0}^{q}\left(t^2+1\right)^{q-i}\left(\sqrt{(t^2-1)^2+4lt}\right)^i+\sum_{i=0}^{q}(-1)^i\left(t^2+1\right)^{q-i}\left(\sqrt{(t^2-1)^2+4lt}\right)^i\right].$$

We note that the alternating sum double the even values the zero out the odd values, thus the above can be rewritten as

$$=\frac{1}{2^{q-1}}\sum_{\text{even } i}\binom{q}{i}\left(t^2+1\right)^i\left((t^2-1)^2-4lt\right)^{\frac{q-i}{2}}.$$

and by applying it again, on the second multiplicative term,

$$=\frac{1}{2^{q-1}}\sum_{\text{even } i}\binom{q}{i}\left(t^2+1\right)^i\sum_{j=0}^{\frac{q-i}{2}}\binom{\frac{q-i}{2}}{j}\left(t^2-1\right)^{2j}\cdot(4t)^{\frac{q-i}{2}-j}\cdot l^{\frac{q-i}{2}-j}.$$

Combining the two insights results in

$$(6.2)=\frac{1}{2^{t-1}}\left[t^p+1+\sum_{l=1}^{t}(-1)^l\left(\frac{1}{2^{q-1}}\sum_{\text{even } i}\binom{q}{i}(t^2+1)^i\sum_{j=0}^{\frac{q-i}{2}}\binom{\frac{q-i}{2}}{j}(t^2-1)^{2j}\,(4tl)^{\frac{q-i}{2}-j}\,l^{\frac{q-i}{2}-j}\right)\right].$$

We further note that for $l = 0$, the term in the inner parentheses is non-zero only when $\frac{q-i}{2} = j$. In this case we get, using the binomial theorem once more,

$$\frac{1}{2^{q-1}}\sum_{\text{even } i}\binom{q}{i}(t^2+1)^i(t^2-1)^{q-i}=\left(\frac{t^2+1+t^2-1}{2}\right)^q+\left(\frac{t^2+1-t^2+1}{2}\right)^q=1+t^p.$$

Therefore, we can rewrite (6.2) as

$$(6.2)=\frac{1}{2^{t-1}}\left(\sum_{l=0}^{t}(-1)^l\frac{1}{2^{q-1}}\sum_{\text{even } i}\binom{q}{i}(t+1)^i\sum_{j=0}^{\frac{q-i}{2}}\binom{\frac{q-i}{2}}{j}(t-1)^{2j}4^{\frac{q-i}{2}-j}l^{\frac{q-i}{2}-j}\right)$$

and using [LW16a] observation,

$$=\frac{1}{2^{t-1}}(-1)^t t!\sum_{\text{even } i}\binom{q}{i}(t+1)^i\sum_{j=0}^{\frac{q-i}{2}}\binom{\frac{q-i}{2}}{j}(t-1)^{2j}4^{\frac{q-i}{2}-j}\left\{{\frac{q-i}{2} \atop t}\right\}$$

where $\left\{{\frac{q-i}{2} \atop t}\right\}$ are Stirling numbers of the second kind. As for a fixed t all terms are of the same sign, the sum vanishes only when $\left\{{\frac{q-i}{2} \atop t}\right\} = 0\ \forall i$, which happens when $t > q/2 = p/4$. □

We remark that Lemma 6.2 extends to $p \equiv 2 \pmod 4$ when $t \le (p-2)/4$, by replacing in the proof $q = p/2$ with $\tilde{q} = (p-2)/2$. The next theorem follows easily by combining Theorem 6.1 and Lemma 6.2.

Theorem 1.2. *For every $p \in 2\mathbb{Z}_{\ge 2}$ there is $\varepsilon(p) > 0$ such that every algorithm that makes one pass over an $O_p(1)$-sparse matrix $A \in \mathbb{R}^{n\times n}$ streamed in row-order, and then outputs a $(1 \pm \varepsilon(p))$-approximation to $\|A\|_{S_p}^p$ with probability at least $2/3$, must use $\Omega(n^{1-4/\lfloor p \rfloor_4})$ bits of space.*

Proof. Let us first assume that $p \equiv 0 \pmod 4$. As shown in Lemma 6.2, the gap condition (6.1) holds for $f(X) = \|X\|_{S_p}^p$ and $t = p/4$, thus by Theorem 6.1 the space complexity is $\Omega(n^{1-1/t}) = \Omega(n^{1-4/p})$ bits. For $p \equiv 2 \pmod 4$ the claim holds for $t = (p-2)/4$, yielding an $\Omega(n^{1-4/(p-2)})$ bits lower bound. □

We note that for $p \equiv 0 \pmod 4$ the above matches up to logarithmic factors the upper bound for the row-order algorithm presented in [BCK+18], i.e. tight for matrices in which every row and column has $O(1)$ non-zero elements. For $p \equiv 2 \pmod 4$, there is a small gap: the lower bound is $\Omega(n^{1-4/(p-2)})$ while the upper bound obtained in [BCK+18] is $\tilde{O}_k(n^{1-4/(p+2)})$.

7 $O_\varepsilon(1)$-Space Algorithm for Schatten 4-Norm of General Matrices

We present an $O(1/\varepsilon^2)$-space algorithm for $(1+\varepsilon)$-approximation of the Schatten 4-norm in the row-order model. As this result does not depend on the sparsity and is applicable to any matrix, it significantly improves the previously known row-order algorithm, presented in [BCK+18] that uses space $\tilde{O}_{p,\varepsilon}(k)$, and is also better than the result of Section 4.1.

The algorithm exploits the fact that $A^\top A = \sum_i a_i^\top a_i$ (i.e. summing over the outer product of every row with itself), to sketch the Frobenius norm $\sum_{j_1,j_2}((A^\top A)_{j_1,j_2})^2 = \|A^\top A\|_F^2 = \|A\|_{S_4}^4$. To do so, it uses two random 4-wise independent vectors, following an idea presented in [IM08] (extending the classic [AMS99] result), as follows.

Lemma 7.1 (Lemma 3.1 in [IM08])**.** *Consider random $h, g \in \{-1,1\}^n$ where each vector is 4-wise independent (and independent of the other one). Let $v \in \mathbb{R}^{n^2}$ and $z_j = h_{j_1} g_{j_2}$ for $j \in [n]^2$, and define $\Upsilon = (\sum_{j\in[n]^2} z_j v_j)^2$. Then*

$$\mathbb{E}[\Upsilon] = \sum_{j\in[n]^2} v_j^2, \quad \textit{and} \quad Var(\Upsilon) \le 3(\mathbb{E}[\Upsilon])^2.$$

Algorithm 5 Algorithm for Schatten 4-Norm of General Matrices in Row-Order Streams

Input: $A \in \mathbb{R}^{n\times n}$ streamed in row-order, $\varepsilon > 0$.

```
in parallel $m = \tilde{O}(1/\varepsilon^2)$ times do
    init: $Y \leftarrow 0$ and choose two random 4-wise independent vectors $h, g \in \{-1,1\}^n$
    upon receiving row $a_i$, update: $Y \mathrel{+}= \langle h, a_i\rangle\langle g, a_i\rangle$
    let $\Upsilon \leftarrow Y^2$
return average of the $m$ copies of $\Upsilon$
```

Theorem 7.2. *Suppose that $A \in \mathbb{R}^{n\times n}$ is a matrix given in one-pass row-order model. Algorithm 5 uses space $O(1/\varepsilon^2)$ and outputs a $(1+\varepsilon)$-approximation to $\|A\|_{S_4}^4$ with probability at least $2/3$.*

Proof. Consider one copy of the independent sketches. Using simple manipulations, we can write:

$$Y = \sum_i \left(\sum_{j_1} h_{j_1} A_{i,j_1} \right) \left(\sum_{j_2} g_{j_2} A_{i,j_2} \right) = \sum_{j_1,j_2} h_{j_1} g_{j_2} (A^\top A)_{j_1,j_2}$$

By looking at $A^\top A$ as vector of dimension n^2, it easily follows from 7.1 that $\mathbb{E}[\Upsilon] = \|A^\top A\|_F^2 = \|A\|_{S_4}^4$ and $\mathrm{Var}(\Upsilon) \leq 3\|A\|_{S_4}^8$. Repeating the sketch $O(1/\varepsilon^2)$ times independently, decreases the variance and gives the desired result (by Chebyshev's inequality). □

8 Applications

In this section we present two applications of our Schatten-norm algorithms to some common functions of the spectrum, by approximating these functions using low-degree polynomials. We employ the well-known idea that just as functions $f : \mathbb{R} \to \mathbb{R}$ can be approximated in a specific interval by polynomials arising from a Taylor expansion (or using Chebyshev polynomials), spectral functions can be approximated by matrix polynomials if the matrix eigenvalues lie in a bounded range. We just need to implement this method in a space-efficient streaming fashion. In some applications we require the input matrix to have a bounded spectrum. Unfortunately, there is no known streaming algorithm to estimate the spectrum of an input matrix.

8.1 Approximating Spectral Sums of Positive Definite Matrices

We demonstrate how our Schatten-norm estimators can be used to approximate commonly used spectral functions of sparse matrices presented as a data stream. We consider three different spectral functions, log-determinant, trace of matrix inverse and Estrada index of a Laplacian matrix, that all belong to the class of spectral sums, as defined below. These results apply to sparse matrices that are either positive definite (PD), positive semidefinite (PSD), or Laplacian. Throughout, $\log x$ denotes the natural logarithm of x.

Definition 8.1 (Spectral Sums [HMAS17])**.** Given a function $f : \mathbb{R} \to \mathbb{R}$ and a matrix $A \in \mathbb{R}^{n \times n}$ with real eigenvalues $\lambda_1, \ldots, \lambda_n$, a spectral sum is defined as

$$S_f(A) = \mathsf{Tr}(f(A)) = \sum_{i=1}^{n} f(\lambda_i).$$

When $f(x) = \log x$, the sum is known as log-determinant, when $f(x) = 1/x$ it is known as the trace of the matrix inverse, and when $f(x) = \exp(x)$ it is known as Estrada index.

Theorem 8.2. *For every spectral function S_f from the table below, there is an algorithm with the following guarantee. Given as input $\varepsilon, \theta > 0$, and a k-sparse matrix $A \in \mathbb{R}^{n \times n}$ presented as a row-order stream and whose eigenvalues all lie in the interval I_f given in the table, the algorithm makes $\lfloor m_f/4 \rfloor + 1$ passes over the stream using W_f words of space and outputs an estimate $\rho(A)$ such that*

$$\Pr\left[\rho(A) \in (1 \pm 2\varepsilon) S_f(A)\right] \geq 2/3.$$

$\boldsymbol{S_f}$	$\boldsymbol{I_f}$	$\boldsymbol{m_f}$	$\boldsymbol{W_f}$
Spectral Function	*Spectrum Interval*		*Words of Space*
Log-Determinant	$[\theta, 2)$	$\lceil \frac{1}{\theta} \cdot \log \frac{1}{\varepsilon} \rceil$	$O_{m_f}(\varepsilon^{-2} k^{3m_f/2-3})$
Trace of Matrix Inverse	$[\theta, 2)$	$\lceil \frac{1}{\theta} \cdot \log \frac{1}{\varepsilon} \rceil$	$O_{m_f}(\varepsilon^{-2} k^{3m_f/2-3})$
Estrada Index of a Laplacian	$[0, \theta]$ [7]	$\lceil (e\theta + 1) \log \frac{1}{\varepsilon} - 1 \rceil$	$O_{m_f}(\varepsilon^{-2} k^{m_f/2})$

At a high level, the proof follows that of Boutsidis et al. [BDK+17], who present a time-efficient algorithm for approximating the log-determinant of PD matrices. Besides extending their method to two other spectral sums, the main difference is that we need to adapt their argument to be space-efficient in the streaming model. More specifically, the log-determinant of a PD matrix is approximated in [BDK+17, Lemma 7] by a truncated version (i.e., only the first summands) of its Taylor expansion

$$\log \det(A) = -\sum_{p=1}^{\infty} \mathsf{Tr}((I_n - A)^p)/p. \tag{8.1}$$

They then approximate the required matrix traces by multiplying the respective matrix by a Gaussian vector (from left and right), which can be implemented faster than matrix powering, by starting with the vector and repeatedly multiply it by a matrix. While this is time-efficient, it is not space-efficient when the input matrix is sparse, in which case our streaming algorithm has better space complexity. One other difference to note is that our algorithm approximates each of the above-mentioned traces *separately*, and thus we need all the Taylor expansion coefficients to be non-negative, which indeed applies for these three spectral functions.[8]

Proof. We follow the proof framework of Lemma 8 in [BDK+17], achieving the desired relative error of the desired spectral function using a truncated version of its Taylor expansion, consisting m_f terms. The first relative error is due to the tail of the series, i.e. the terms that were not considered in the final estimate. For the log-determinant, the above-mentioned Lemma 8 shows that it suffices to $(1 \pm \varepsilon)$-approximate the first $m_f = \lceil \frac{1}{\theta} \cdot \log \frac{1}{\varepsilon} \rceil$ terms of its Taylor expansion (8.1) in order to obtain a $(1 \pm 2\varepsilon)$-approximation of $\log \det(A)$. The same proof holds also for the Taylor expansion

$$\mathsf{Tr}(A^{-1}) = \sum_{p=1}^{\infty} -\mathsf{Tr}((I_n - A)^p)$$

and obtaining a $(1 \pm 2\varepsilon)$-approximation of $\mathsf{Tr}(A^{-1})$ (for the same value of m_f). To achieve this error bound for the Estrada index of a Laplacian, the number of values of its Taylor series (see e.g. [DL11, GDR11])

$$\mathsf{Tr}(\exp(A)) = \sum_{p=0}^{\infty} \mathsf{Tr}(A^p)/p! \tag{8.2}$$

that need to be approximated is $m_f = \lceil (e\theta + 1) \log \frac{1}{\varepsilon} - 1 \rceil$, as shown in Appendix A.4.

We are left to explain how to $(1 \pm \varepsilon)$-approximate the first m_f values of the Taylor expansion (causing the other relative error). Recall that if a matrix B is PSD then $\mathsf{Tr}(B^p) = \sum \lambda_i^p = \|B\|_{S_p}^p$, where $\lambda_1, \ldots, \lambda_n \geq 0$ are its eigenvalues. Furthermore, for such matrices our algorithm works for every integer $p \geq 2$, and therefore this relative error is immediate from Theorems 1.1 and 4.1.

To conclude, we can compute the traces of B^p in parallel for all integer $p = 2, 3, \ldots, m_f$ using Algorithm 1, while for $p = 1$ one can compute $\|B\|_{S_1}^1 = \mathsf{Tr}(B)$ by directly summing the main

[7] If the underlying graph is unweighted then the largest eigenvalue is bounded by the degree, i.e. $\theta \leq 2k$.

[8] Our method extends to alternating Taylor sums if the coefficients decrease by a constant factor, by bounding the approximation error difference of every two consecutive summands. One such an example is $\mathsf{Tr}(\exp(-A))$.

diagonal entries. These parallel executions take $\lfloor m/4 \rfloor + 1$ passes and the total space is at most $O_m(\varepsilon^{-2}k^{3m/2-3})$ words of space for log-determinant and trace of matrix inverse, and $O_m(\varepsilon^{-2}k^{m/2})$ words of space for the Estrada index of a Laplacian matrix. □

8.2 Approximating the Spectrum of PSD matrices

We present an application of our algorithm to (weakly) estimate the spectrum of a matrix, with eigenvalues bounded in $[0, 1]$ using approximations of a "few" Schatten norms of the matrix. This is based on the work of Cohen-Steiner et al. [CKSV18] on approximating the spectrum of a graph which is in turn based on insightful work by Wong and Valiant [KV16] on approximately recovering a distribution from its moments using the Moment Inverse method.

Fix a PSD matrix $A \in \mathbb{R}^{n\times n}$ with eigenvalues $1 \geq \lambda_1 \geq \ldots \geq \lambda_n$ and define the l-th moment of the spectrum to be $\frac{1}{n}\|A\|_{S_l}^l = \frac{1}{n}\sum_{i\in[n]} \lambda_i^l$. Cohen-Steiner et al. show that estimating $O(1/\varepsilon)$ moments of A up to multiplicative error $O(\varepsilon)$ is sufficient to estimate the spectrum of A within earth-mover distance $O(\varepsilon)$. It is well-known that the the L_1 distance between two sorted vectors of length n is exactly n times the earth-mover distance between the corresponding point-mass distributions (uniform probability on each of the n indices). Hence, the recovery scheme of Cohen-Steiner et al. allows us to recover the spectrum within L_1 distance $O(\varepsilon n)$ by estimating only $O\left(\frac{1}{\varepsilon}\right)$ moments of the matrix A. Specifically, we get the following result,

Theorem 8.3 (Theorem 7 in [CKSV18])**.** *Given a constant $\varepsilon > 0$, there exists a parameter $s = \frac{C}{\varepsilon}$ (where $C > 0$ is an absolute constant) and an algorithm R such that, for a PSD matrix $A \in \mathbb{R}^{n\times n}$ with eigenvalues $\lambda = (\lambda_1, \ldots, \lambda_n) \in [0, 1]^n$ and a vector $y \in \mathbb{R}^s$ with the property that $y_i = \|\lambda\|_i^i \pm \exp(-C'\varepsilon)$ for all $i \in [s]$ and absolute constant $C' > 0$, R reads y and outputs a vector $\hat{\lambda}$ such that $\|\lambda - \hat{\lambda}\|_1 \leq \varepsilon n$.*

For an error parameter $\varepsilon > 0$ and parameter $s = \frac{C}{\varepsilon}$ (where $C > 0$ is an absolute constant) as defined in the above theorem, given a k-sparse PSD matrix $A \in \mathbb{R}^{n\times n}$ that is streamed in row-order and whose eigenvalues are in the range $[0, 1]$, one can use Algorithm 1 to compute the vector $y \in \mathbb{R}^s$ with the desired guarantee using space $O(k^{3s/2-3}\exp(-C'\varepsilon))$ for some absolute constant $C' > 0$ and using $\lfloor s/4 \rfloor + 1$ passes over the stream.

9 Experiments

In this section we present numerical experiments illustrating the performance of the row-order Schatten p-norm estimator described in Section 4.1. We simulate the row-order stream by reading the input matrix row by row. The results not only follow theoretical space bounds, showing that the algorithm is indeed independent of the matrix size, but are actually several orders of magnitude better. In addition, the experiments show that the algorithm is robust to noise, and these two results suggest that real-life behavior of the algorithm is significantly better than our theoretical bounds.

The inputs used are $\{0, 1\}^{n\times n}$ matrices, representing collaboration network graphs (nodes represent scientists and edges represent co-authoring a paper) from the e-print arXiv for scientific collaborations in five different areas in Physics. The data was obtained from the Stanford Large Network Dataset Collection [LK14] which was in-turn obtained from [LKF07]. In order to study the effect of sparsity, we "sparsify" each (of five) matrix by sampling 10 nonzero entries in each row uniformly at random (note that max column-sparsity can be larger than 10).

In the first experiment, we use the arXiv General Relativity and Quantum Cosmology collaboration network which has $n = 5242$ rows and columns; after "sparsifying" the matrix as mentioned, the max column-sparsity is 37 and the average sparsity is 6.1. We fix the value of p to be 6, and using our algorithm from Section 4.1, we vary number of estimators (walks) t and compute the *relative error* of the average of the t walks. We repeat this process 10 times for every value of t and plot the mean and standard deviation in Figure 1. In addition, we show in this figure the results of running the same experiment on a "noisy" version of the matrix, by adding to it an error matrix where 1/5 of the entries are drawn independently from $\mathcal{N}(0, 0.1^2)$[9].

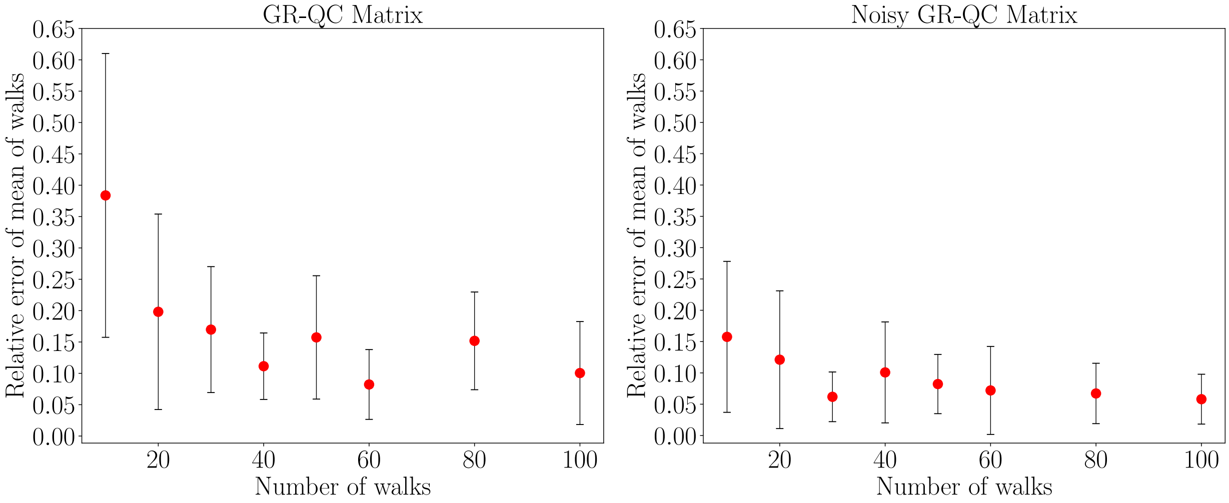

Figure 1: Relative error of Algorithm 1 for Schatten 6-norm of arXiv General Relativity and Quantum Cosmology Collaboration Network: Vary number of walks and plot relative error of the mean of the walks.

Recall that the number of independent walks (estimators) is ultimately the space required by Algorithm 1 (upto the space needed to store a row), as they are run the in parallel. Therefore, the left graph shows that the space actually needed to approximate the Schatten 6-norm of the selected input matrix is significantly smaller than the theoretical bound of Theorem 4.1, which is $O_p\varepsilon^{-2}k(p/2) \approx 135000$. The other graph shows that the algorithm is robust to small random noise, i.e. it works also for nearly-sparse, where every row and column are dominated by a small amount of entries.

In the second experiment, we use all five collaboration networks – General Relativity and Quantum Cosmology ($n = 5242$), High Energy Physics - Phenomenology ($n = 9877$), High Energy Physics - Theory ($n = 12008$), Astro Physics ($n = 18772$) and Condensed Matter ($n = 23133$). For each matrix we compute walks (estimator from Section 4.1) until the mean of the walks is within 10% of the true Schatten 6-norm of the matrix. We repeat this process 10 times for each matrix and plot the median, the first and third quartile of the number of walks for the 10 trials in Figure 2.Since in the second and third experiments, most of the outputs of the 10 trials are concentrated around the median except for very few trials (one or two) which are very large outliers. Hence, we chose to output the first and third quartiles indicating the output of the majority of the trials.

[9] This value assures the l_2-norm of the error in a row is "comparable" to the l_2-norm of the data: $(0.1)^2 \times 5242 \times 0.2 \approx 10 =$ max row-sparsity.

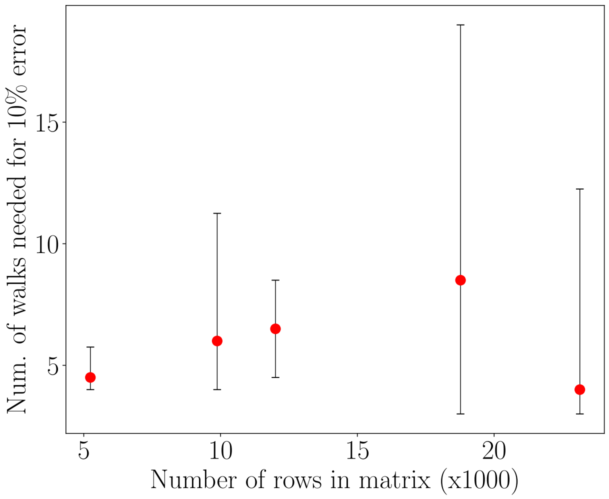

Figure 2: Number of walks to (1 ± 0.1)-approximate Schatten 6-norm of 5 different matrices from arXiv Physics Collaboration Network.

The above figure shows that indeed calculating the space needed to approximate the Schatten norms using our algorithm is independent of the matrix dimension. Again, as in Figure 1, it is easy to see that the number of estimators needed to approximate the Schatten 6-norm of the chosen matrices is several orders of scale better than the theoretical bound.

In our third experiment we compute the number of walks needed for the mean of the walks to be within 10% of the true Schatten p-Norm of the GR-QC matrix for different values of p. We vary the value of p and, for each value of p, compute the number of walks needed for 10 trials and plot the median, first and third quartile of the 10 trials in Figure 3.

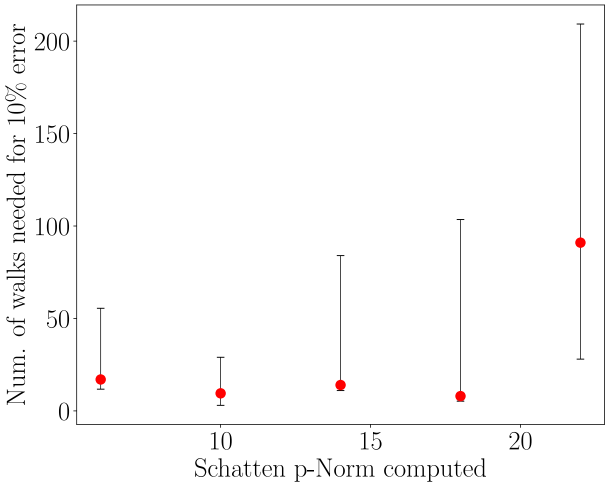

Figure 3: Number of walks to (1 ± 0.1)-approximate Schatten p-norm for arXiv General Relativity and Quantum Cosmology Collaboration Network (GR-QC) for different values of $p \in 2\mathbb{Z}^+$.

The last figure follows the previous figures, and shows that again the numerical results are much better than the theoretical bounds, in this case in the dependence on p. Although there is an expected increase in space as p grows, it is not rapid, and in particular is not exponential. This means, for example, that the space needed to approximate other spectral functions, as explained in Section 8, would be small, suggesting that our algorithm would be practical for such tasks.

References

[AMS99] N. Alon, Y. Matias, and M. Szegedy. The space complexity of approximating the frequency moments. *Journal of Computer and System Sciences*, 58(1):137–147, 1999. doi:10.1006/jcss.1997.1545.

[AN13] A. Andoni and H. L. Nguyen. Eigenvalues of a matrix in the streaming model. In *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2013*, pages 1729–1737, 2013. doi:10.1137/1.9781611973105.124.

[BCK+18] V. Braverman, S. R. Chestnut, R. Krauthgamer, Y. Li, D. P. Woodruff, and L. Yang. Matrix norms in data streams: Faster, multi-pass and row-order. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pages 648–657, 2018. Available from: http://proceedings.mlr.press/v80/braverman18a.html.

[BDK+17] C. Boutsidis, P. Drineas, P. Kambadur, E.-M. Kontopoulou, and A. Zouzias. A randomized algorithm for approximating the log determinant of a symmetric positive definite matrix. *Linear Algebra and its Applications*, 533:95–117, 2017.

[BOV15] V. Braverman, R. Ostrovsky, and G. Vorsanger. Weighted sampling without replacement from data streams. *Information Processing Letters*, 115(12):923 – 926, 2015. doi:10.1016/j.ipl.2015.07.007.

[BS15] M. Bury and C. Schwiegelshohn. Sublinear estimation of weighted matchings in dynamic data streams. In *23rd Annual European Symposium*, ESA'15, pages 263–274, 2015. doi:10.1007/978-3-662-48350-3_23.

[CCF04] M. Charikar, K. C. Chen, and M. Farach-Colton. Finding frequent items in data streams. *Theor. Comput. Sci.*, 312(1):3–15, 2004. doi:10.1016/S0304-3975(03)00400-6.

[Che16] J. Chen. How accurately should I compute implicit matrix-vector products when applying the Hutchinson trace estimator? *SIAM J. Scientific Computing*, 38(6), 2016. doi:10.1137/15M1051506.

[CJ19] G. Cormode and H. Jowhari. Lp samplers and their applications: A survey. *ACM Comput. Surv.*, 52(1):16:1–16:31, 2019. doi:10.1145/3297715.

[CKSV18] D. Cohen-Steiner, W. Kong, C. Sohler, and G. Valiant. Approximating the spectrum of a graph. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '18, pages 1263–1271. ACM, 2018. doi:10.1145/3219819.3220119.

[CW09] K. L. Clarkson and D. P. Woodruff. Numerical linear algebra in the streaming model. In *Proceedings of the Forty-first Annual ACM Symposium on Theory of Computing*, STOC '09, pages 205–214, New York, NY, USA, 2009. ACM. doi:10.1145/1536414.1536445.

[DL11] Z. Du and Z. Liu. On the Estrada and Laplacian Estrada indices of graphs. *Linear Algebra and its Applications*, 435(8):2065–2076, 2011.

[GDR11] I. Gutman, H. Deng, and S. Radenkovic. The Estrada index: An updated survey. *Selected Topics on Applications of Graph Spectra, Math. Inst., Beograd*, pages 155–174, 2011.

[GJP$^+$12] N. Goyette, P. Jodoin, F. Porikli, J. Konrad, and P. Ishwar. Changedetection.net: A new change detection benchmark dataset. In *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–8, June 2012. doi:10.1109/CVPRW.2012.6238919.

[GM99] P. B. Gibbons and Y. Matias. Synopsis data structures for massive data sets. In *External Memory Algorithms*, volume 50 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*. American Mathematical Society, 1999. Available from: http://theory.stanford.edu/~matias/papers/synopsis.pdf.

[Gro09] A. Gronemeier. Asymptotically optimal lower bounds on the NIH-multi-party information complexity of the AND-function and Disjointness. In *26th International Symposium on Theoretical Aspects of Computer Science STACS 2009*, volume 3, pages 505–516. IBFI Schloss Dagstuhl, 2009. doi:10.4230/lipiCS.stacs.2009.1846.

[GvDCB13] J. Ganitkevitch, B. van Durme, and C. Callison-Burch. PPDB: The paraphrase database. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 758–764. Association for Computational Linguistics, 2013. Available from: https://www.aclweb.org/anthology/N13-1092.

[HJ85] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge university press Cambridge, Cambridge, 1985.

[HMAS17] I. Han, D. Malioutov, H. Avron, and J. Shin. Approximating spectral sums of large-scale matrices using stochastic Chebyshev approximations. *SIAM J. Scientific Computing*, 39(4), 2017. doi:10.1137/16M1078148.

[HMS15] I. Han, D. Malioutov, and J. Shin. Large-scale log-determinant computation through stochastic Chebyshev expansions. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, pages 908–917, 2015.

[IM08] P. Indyk and A. McGregor. Declaring independence via the sketching of sketches. In *Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 737–745. SIAM, 2008. Available from: http://dl.acm.org/citation.cfm?id=1347082.1347163.

[Jay09] T. S. Jayram. Hellinger strikes back: A note on the multi-party information complexity of AND. In *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*, pages 562–573, 2009. doi:10.1007/978-3-642-03685-9_42.

[JW09] T. S. Jayram and D. P. Woodruff. The data stream space complexity of cascaded norms. In *50th Annual IEEE Symposium on Foundations of Computer Science*, pages 765–774, 2009. doi:10.1109/FOCS.2009.82.

[KO19] A. Khetan and S. Oh. Spectrum estimation from a few entries. *Journal of Machine Learning Research*, 20(21):1–55, 2019. Available from: http://jmlr.org/papers/v20/18-027.html.

[KV16] W. Kong and G. Valiant. Spectrum estimation from samples. *The Annals of Statistics*, 45, 01 2016. doi:10.1214/16-AOS1525.

[Lib13] E. Liberty. Simple and deterministic matrix sketching. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '13, pages 581–588. ACM, 2013. doi:10.1145/2487575.2487623.

[LK14] J. Leskovec and A. Krevl. SNAP Datasets: Stanford large network dataset collection. http://snap.stanford.edu/data, June 2014.

[LKF07] J. Leskovec, J. Kleinberg, and C. Faloutsos. Graph evolution: Densification and shrinking diameters. *ACM Trans. Knowl. Discov. Data*, 1(1):2?es, March 2007. Available from: https://doi.org/10.1145/1217299.1217301, doi:10.1145/1217299.1217301.

[LNW14] Y. Li, H. L. Nguyen, and D. P. Woodruff. On sketching matrix norms and the top singular vector. In *Proceedings of the 25th annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1562–1581. SIAM, 2014. doi:10.1137/1.9781611973402.114.

[LW16a] Y. Li and D. P. Woodruff. On approximating functions of the singular values in a stream. In *Proceedings of the 48th annual ACM symposium on Theory of Computing (STOC)*, pages 726–739. ACM, 2016. doi:10.1145/2184319.2184343.

[LW16b] Y. Li and D. P. Woodruff. Tight bounds for sketching the operator norm, Schatten norms, and subspace embeddings. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2016*, pages 39:1–39:11, 2016. doi:10.4230/LIPIcs.APPROX-RANDOM.2016.39.

[LW17] Y. Li and D. P. Woodruff. Embeddings of Schatten norms with applications to data streams. In *44th International Colloquium on Automata, Languages, and Programming, ICALP 2017*, pages 60:1–60:14, 2017. doi:10.4230/LIPIcs.ICALP.2017.60.

[LZYQ15] X. Liu, G. Zhao, J. Yao, and C. Qi. Background subtraction based on low-rank and structured sparse decomposition. *IEEE Transactions on Image Processing*, 24(8):2502–2514, Aug 2015. doi:10.1109/TIP.2015.2419084.

[MNS$^+$18] C. Musco, P. Netrapalli, A. Sidford, S. Ubaru, and D. P. Woodruff. Spectrum approximation beyond fast matrix multiplication: Algorithms and hardness. In *9th Innovations in Theoretical Computer Science Conference (ITCS 2018)*, volume 94 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 8:1–8:21, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi:10.4230/LIPIcs.ITCS.2018.8.

[MW10] M. Monemizadeh and D. P. Woodruff. 1 pass relative-error Lp-sampling with applications. In *Proceedings of the Twenty-first Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '10, pages 1143–1160, Philadelphia, PA, USA, 2010. Society for Industrial and Applied Mathematics. Available from: http://dl.acm.org/citation.cfm?id=1873601.1873693.

[NPS16] E. D. Napoli, E. Polizzi, and Y. Saad. Efficient estimation of eigenvalue counts in an interval. *Numerical Lin. Alg. with Applications.*, 23(4):674–692, 2016. doi:10.1002/nla.2048.

[O'D14] R. O'Donnell. *Analysis of Boolean Functions*. Cambridge University Press, New York, NY, USA, 2014.

[Rui96] S. M. Ruiz. An algebraic identity leading to Wilson's theorem. *The Mathematical Gazette*, 80(489):579–582, 1996.

[SS12] W. Schudy and M. Sviridenko. Concentration and moment inequalities for polynomials of independent random variables. In *Proceedings of the Twenty-third Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '12, pages 437–446. SIAM, 2012. Available from: http://dl.acm.org/citation.cfm?id=2095116.2095153.

[UCS17] S. Ubaru, J. Chen, and Y. Saad. Fast estimation of $tr(f(a))$ via stochastic Lanczos quadrature. *SIAM J. Matrix Analysis Applications*, 38(4):1075–1099, 2017. doi:10.1137/16M1104974.

[US17] S. Ubaru and Y. Saad. Applications of trace estimation techniques. In *High Performance Computing in Science and Engineering - Third International Conference, HPCSE 2017, Karolinka, Czech Republic, May 22-25, 2017, Revised Selected Papers*, pages 19–33, 2017. doi:10.1007/978-3-319-97136-0_2.

[Vit85] J. S. Vitter. Random sampling with a reservoir. *ACM Trans. Math. Softw.*, 11(1):37–57, 1985. doi:10.1145/3147.3165.

[Vit01] J. S. Vitter. External memory algorithms and data structures: Dealing with massive data. *ACM Computing Survey.*, 33(2):209–271, 2001. doi:10.1145/384192.384193.

[VY11] E. Verbin and W. Yu. The streaming complexity of cycle counting, sorting by reversals, and other problems. In *Proceedings of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 11–25, 2011. Available from: http://dl.acm.org/citation.cfm?id=2133036.2133038.

[WLK$^+$16] L. Wu, J. Laeuchli, V. Kalantzis, A. Stathopoulos, and E. Gallopoulos. Estimating the trace of the matrix inverse by interpolating from the diagonal of an approximate inverse. *J. Comput. Physics*, 326:828–844, 2016. doi:10.1016/j.jcp.2016.09.001.

[XGL$^+$16] Y. Xie, S. Gu, Y. Liu, W. Zuo, W. Zhang, and L. Zhang. Weighted Schatten p-norm minimization for image denoising and background subtraction. *IEEE Transactions on Image Processing*, 25:4842–4857, 2016.

[ZWJ15] Y. Zhang, M. J. Wainwright, and M. I. Jordan. Distributed estimation of generalized matrix rank: Efficient algorithms and lower bounds. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pages 457–465. JMLR.org, 2015. Available from: http://dl.acm.org/citation.cfm?id=3045118.3045168.

A Appendix

A.1 Proof of Fact 2.1

Let $M = AA^\top$ be a PSD matrix, with eigenvalues $\lambda_1 \geq \ldots \geq \lambda_n \geq 0$. Let $\vec{m}, \vec{\lambda} \in \mathbb{R}^n$ be the vectors corresponding to the diagonal entries of M and the eigenvalues of M respectively, both in non-increasing order. Then, by Schur-Horn theorem (Theorem 4.3.26 in [HJ85]), $\vec{\lambda}$ weakly majorizes $\vec{m}$, i.e. $\sum_{i=1}^r \lambda_i \geq \sum_{i=1}^r m_i$ for all $r \in [n]$.

Since $f(y) = \sum_{i=1}^n y_i^t$ is a Schur-convex function for $y \in \mathbb{R}^n$ and $t \geq 1$, we have that $\sum_{i=1}^n \lambda_i^t \geq \sum_{i=1}^n m_i^t$. The statement follows from the fact that $\sum_{i=1}^n \lambda_i^t = \|AA^\top\|_{S_t}^t = \|A\|_{S_{2t}}^{2t}$ and $m_i = \|a_i\|_2^2$ for all $i \in [n]$.

A.2 Proof of Theorem 2.2

It is easy to see that the estimator is unbiased; $\mathbb{E}\left[\hat{Z}\right] = \sum_{i\in[n]} \frac{z_i}{\tau_i} \cdot \tau_i = z$. Bounding the variance can be done as follows,

$$\mathrm{Var}(\hat{Z}) \leq \mathbb{E}\left[(\hat{Z})^2\right] = \sum_{i\in[n]} \left(\frac{z_i}{\tau_i}\right)^2 \tau_i = \sum_{i\in[n]} \left(\frac{|z_i|}{\tau_i}\right)^2 \tau_i.$$

Since for each $i \in [n]$ we have $\tau_i \geq \frac{|z_i|}{\lambda z}$, we can bound $\mathrm{Var}(\hat{Z}) \leq \sum_{i\in[n]} (\lambda z)^2 \tau_i = (\lambda z)^2$

A.3 Proof of Lemma 5.2

The expectation is straight forward. First assume $r = 1$:

$$\mathbb{E}\left[Y\right] = \mathbb{E}\left[\frac{z_{i,j}}{\tau_i \tau_j}\right] = \sum_{l\in[n], m\in[n]} \frac{z_{l,m}}{\tau_l \tau_m} \tau_l \tau_m = z$$

and then using the linearity of expectation,

$$\mathbb{E}\left[Y\right] = \frac{1}{r^2} \sum_{u\in[r], v\in[r]} \mathbb{E}\left[\frac{z_{i_u, j_v}}{\tau_{i_u} \tau_{j_v}}\right] = \frac{1}{r^2} \sum_{u\in[r], v\in[r]} z = z$$

For the variance,

$$\mathbb{E}Y^2 = \frac{1}{r^4} \sum_{u,v} \left(\mathbb{E}\left[\sum_{\substack{u' \neq u \\ v' \neq v}} \frac{z_{i_u, j_v}}{\tau_{i_u} \tau_{j_v}} \cdot \frac{z_{i_{u'}, j_{v'}}}{\tau_{i_{u'}} \tau_{j_{v'}}} \right] + \mathbb{E}\left[\left(\frac{z_{i_u, j_v}}{\tau_{i_u} \tau_{j_v}} \right)^2 \right] + \mathbb{E}\left[\sum_{u \neq u'} \frac{z_{i_u, j_v}}{\tau_{i_u} \tau_{j_v}} \cdot \frac{z_{i_{u'}, j_v}}{\tau_{i_{u'}} \tau_{j_v}} \right] + \mathbb{E}\left[\sum_{v \neq v'} \frac{z_{i_u, j_v}}{\tau_{i_u} \tau_{j_v}} \cdot \frac{z_{i_u, j_{v'}}}{\tau_{i_u} \tau_{j_{v'}}} \right] \right)$$

As i_u and $i_{u'}$ are independent for $u \neq u'$, and similarly for j_v and $j_{v'}$ for $v \neq v'$, we get

$$\begin{aligned} &= \frac{1}{r^4} \left(r^2(r-1)^2 z^2 + r^2 \sum_{l,m} \frac{z_{l,m}}{\tau_l} \cdot \frac{z_{l,m}}{\tau_m} + r^2(r-1) \sum_{l,m,m'} \frac{z_{l,m'}}{\tau_l} \cdot z_{l,m} + r^2(r-1) \sum_{l,l',m} \frac{z_{l',m}}{\tau_m} \cdot z_{l,m} \right) \\ &\leq z^2 + \frac{1}{r^2} \sum_{l,m} \frac{|z_{l,m}|}{\tau_l \tau_m} \cdot |z_{l,m}| + \frac{1}{r} \sum_{l,m,m'} \frac{|z_{l,m'}|}{\tau_l} \cdot |z_{l,m}| + \frac{1}{r} \sum_{l,l',m} \frac{|z_{l',m}|}{\tau_m} \cdot |z_{l,m}| \end{aligned}$$

As the first term is just $(\mathbb{E}[Y])^2$, it holds that

$$\mathrm{Var}(Y) \leq \frac{1}{r^2} \sum_{l,m \in N(l)} \frac{|z_{l,m}|}{\tau_l \tau_m} \cdot |z_{l,m}| + \frac{1}{r} \sum_{l,m,m' \in N(l)} \frac{|z_{l,m'}|}{\tau_l} \cdot |z_{l,m}| + \frac{1}{r} \sum_{m,l \in N(m), l' \in N(m)} \frac{|z_{l',m}|}{\tau_m} \cdot |z_{l,m}|$$

Recalling that $z_{l,m} = 0$ for all $(l,m) \notin E$, we can rewrite the above as

$$= \frac{1}{r^2} \sum_{l,m \in N(l)} \frac{|z_{l,m}|}{\tau_l \tau_m} \cdot |z_{l,m}| + \frac{1}{r} \sum_{l,m \in N(l), m' \in N(l)} \frac{|z_{l,m'}|}{\tau_l} \cdot |z_{l,m}| + \frac{1}{r} \sum_{m,l \in N(m), l' \in N(m)} \frac{|z_{l',m}|}{\tau_m} \cdot |z_{l,m}|$$

and using the bound on the probability,

$$\leq \frac{\lambda^2 z}{r^2} \sum_{l,m \in N(l)} |z_{l,m}| + \frac{\lambda z}{r} \sum_{l,m \in N(l), m' \in N(l)} |z_{l,m}| + \frac{\lambda z}{r} \sum_{m,l \in N(m), l' \in N(m)} |z_{l,m}|$$

Finally, using the bounds on maximum degrees, we get

$$\leq \left(\frac{\lambda^2}{r^2} + \frac{2\lambda\Delta}{r} \right) z \sum_{i,j \in [n]} |z_{i,j}|$$

A.4 Bounding the tail of the Estrada index Taylor expansion (Theorem 8.2)

We bound the tail of the Estrada index Taylor expansion (8.2), i.e. $\left|\sum_{p=m+1}^{\infty} \frac{\mathsf{Tr}(A^p)}{p!}\right| \leq \varepsilon \left|\mathsf{Tr}(\exp(A))\right|$ for $m = \lceil (e\theta + 1)\log(1/\varepsilon) - 1 \rceil$.

$$\left| \sum_{p=m+1}^{\infty} \frac{\mathsf{Tr}(A^p)}{p!} \right| \leq \left| \sum_{p=m+1}^{\infty} \frac{\mathsf{Tr}(A^{m+1}A^{p-(m+1)})}{(m+1)!(p-(m+1))!} \right| .$$

Using $\mathsf{Tr}(AB) \leq \|A\|_{S_\infty} \cdot \mathsf{Tr}(B)$ which follows from Von Neuman's trace inequality (see [BDK+17]),

$$\leq \frac{\|A^{m+1}\|_{S_\infty}}{(m+1)!} \left| \sum_{p=m+1}^{\infty} \frac{\mathsf{Tr}(A^{p-(m+1)})}{(p-(m+1))!} \right| ,$$

and by the bound on the largest eigenvalue and Stirling's formula,

$$\leq \frac{(e\theta)^{m+1}}{(m+1)^{m+3/2}\sqrt{2\pi}} \left| \sum_{p=0}^{\infty} \frac{\mathsf{Tr}(A^p)}{p!} \right|$$

$$\leq \left(\frac{e\theta}{m+1} \right)^{m+1} \left|\mathsf{Tr}(\exp(A))\right|$$

Setting $m = \lceil (e\theta + 1)\log(1/\varepsilon) - 1 \rceil$ and using $(1 - x^{-1})^x \leq e^{-1}$ (for $x > 0$) guarantees that

$$\left(\frac{e\theta}{m+1} \right)^{m+1} \leq \left(\frac{e\theta}{(e\theta+1)\log(1/\varepsilon)} \right)^{m+1} \leq \left(1 - \frac{e\theta}{e\theta+1} \right)^{(e\theta+1)\log(1/\varepsilon)} = \varepsilon.$$