2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)

High-Dimensional Geometric Streaming in Polynomial Space

David P. Woodruff
Carnegie Mellon University
dwoodruf@cs.cmu.edu

Taisuke Yasuda
Carnegie Mellon University
taisukey@cs.cmu.edu

***Abstract*—Many existing algorithms for streaming geometric data analysis have been plagued by exponential dependencies in the space complexity, which are undesirable for processing high-dimensional data sets, i.e., large d. In particular, once $d \geq \log n$, there are no known non-trivial streaming algorithms for problems such as maintaining convex hulls and Löwner–John ellipsoids of n points, despite a long line of work in high-dimensional streaming computational geometry since [2].**

We simultaneously improve all of these results to $\mathrm{poly}(d, \log n)$ bits of space by trading off with a $\mathrm{poly}(d, \log n)$ factor distortion. We achieve these results in a unified manner, by designing the first streaming algorithm for maintaining a coreset for ℓ_∞ subspace embeddings with $\mathrm{poly}(d, \log n)$ space and $\mathrm{poly}(d, \log n)$ distortion. Our algorithm also gives similar guarantees in the *online coreset* model. Along the way, we sharpen known results for online numerical linear algebra by replacing a log condition number dependence with a $\log n$ dependence, answering an open question of [13]. Our techniques provide a novel connection between leverage scores, a fundamental object in numerical linear algebra, and computational geometry.

For ℓ_p subspace embeddings, our improvements in online numerical linear algebra yield nearly optimal trade-offs between space and distortion for one-pass streaming algorithms. For instance, we obtain a deterministic coreset using $O(d^2 \log n)$ space and $O((d \log n)^{\frac{1}{2}-\frac{1}{p}})$ distortion for $p > 2$, whereas previous deterministic algorithms incurred a $\mathrm{poly}(n)$ factor in the space or the distortion [26].

Our techniques have implications also in the offline setting, where we give optimal trade-offs between the space complexity and distortion of a subspace sketch data structure, which preprocesses an $n \times d$ matrix $\mathbf{A}$ and outputs $\|\mathbf{Ax}\|_p$ up to a $\mathrm{poly}(d)$ factor distortion for any $\mathbf{x}$. To do this we give an elementary proof of a "change of density" theorem of [42] and make it algorithmic.[1]

***Index Terms*—computational geometry, streaming**

I. Introduction

Data science has permeated modern computer science in the last few decades, leading to a surge in demand for geometric data processing algorithms on large data sets. Two decades ago, the data sets studied in practice, represented by an $n \times d$ matrix $\mathbf{A}$, had many rows (large n) and small dimension ($d = O(1)$). Driven by such applications, many *streaming algorithms* were developed, which only require one or a few passes through a stream which allows access to the rows $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n \in \mathbb{R}^d$ one at a time. In this setting, ε-*kernels* were introduced by [2], [3], which gave a unified approach towards obtaining $(1+\varepsilon)$-factor approximations using $\varepsilon^{-\Theta(d)}$ space for a wide range of geometric problems, including width, convex hull, and minimum enclosing spherical shell, to name just a few of the applications of ε-kernels.

Since then, the dimensionality of data sets encountered in practice has increased dramatically, and space complexities that scale exponentially in d, or even a large polynomial (say d^4), can no longer be considered practical. Some geometric problems have adapted to this high-dimensional setting, including minimum enclosing cylinder [15, Theorem 3.1], minimum enclosing ball (MEB) [5], [17], [61], and diameter [5, Theorem 3.2], by tolerating a larger $O(1)$-factor distortion. [5] also give lower bounds for the MEB and diameter problems, showing that any one-pass streaming algorithm with less than an α-factor distortion must use $\exp(\mathrm{poly}(d))$ bits of space, where $\alpha = \frac{1+\sqrt{2}}{2}$ for MEB and $\alpha = \sqrt{2}$ for diameter. Furthermore, [5] show that the width problem requires $\exp(\mathrm{poly}(d))$ bits of space for any algorithm achieving distortion smaller than $d^{1/3}/8$. Thus, distortions at least $\mathrm{poly}(d)$ are necessary for some of these problems to achieve $\mathrm{poly}(d)$ bits of space. However, many problems still do not have polynomial space algorithms, even with $\mathrm{poly}(d)$ distortions, such as computing width, Löwner–John ellipsoids [5], [51], ℓ_p subspace embeddings for large p [26], and convex hulls [12].

A. Our Contributions

In this work, we address the lack of streaming algorithms for geometric problems in the high-dimensional setting by providing a unified approach towards achieving $\mathrm{poly}(d, \log n)$ space and distortion. As argued before, a dependence of $\mathrm{poly}(d)$ in the distortion is necessary for polynomial space algorithms, and is arguably natural since many geometric summarization problems

David P. Woodruff and Taisuke Yasuda were supported by ONR grant N00014-18-1-2562 and a Simons Investigator Award.

[1]Extended abstract; full version available at https://arxiv.org/abs/2204.03790.

2575-8454/22/$31.00 ©2022 IEEE
DOI 10.1109/FOCS54457.2022.00075

inherently incur such losses in the distortion, e.g., for Löwner–John ellipsoids.

To obtain our results for streaming geometry, we design the first one-pass streaming algorithm for the ℓ_∞ *subspace sketch problem*. That is, given a row arrival stream for $\mathbf{A} \in \mathbb{Z}^{n\times d}$ with entries bounded by $\mathrm{poly}(n)$, we show how to maintain a coreset $S \subseteq [n]$ of size at most $|S| \le O(d\log n)$ such that for all $\mathbf{x} \in \mathbb{R}^d$,

$$\|\mathbf{A}|_S\mathbf{x}\|_\infty \le \|\mathbf{A}\mathbf{x}\|_\infty \le O(\sqrt{d\log n})\|\mathbf{A}|_S\mathbf{x}\|_\infty.$$

Our algorithm is deterministic and uses only $O(d^2\log^2 n)$ bits of space, which is an optimal trade-off between the space complexity and distortion, up to $\mathrm{poly}\log n$ factors. In fact, our algorithm has the property that each $i \in S$ is selected *irrevocably*, i.e., we immediately decide whether to permanently keep or discard the row $\mathbf{a}_i$. Such algorithms can be considered under the *online coreset* model, in which the input matrix $\mathbf{A} \in \mathbb{R}^{n\times d}$ is now allowed to take real values, and the algorithm's complexity is measured by the number of rows it stores. Under this model, our algorithm stores $O(d\log(n\kappa^{\mathsf{OL}}))$ rows and achieves a distortion of $O(\sqrt{d\log(n\kappa^{\mathsf{OL}})})$, where $\kappa^{\mathsf{OL}} = \|\mathbf{A}\|_2 \max_{i=1}^n \|\mathbf{A}_i^-\|_2{}^2$ is the *online condition number of* $\mathbf{A}$ [13]. Various linear algebraic and geometric problems have been considered in the online model, including spectral approximation [24], low rank approximation [11], [13], and ℓ_1 subspace embeddings [13].

Note that the ℓ_∞ subspace sketch problem is of central importance in computational geometry: it is closely related to directional width estimation [2], [3] as well as the polytope membership problem [8]. It can also be used to approximate maximum inner product search, for which sampling-based algorithms have recently received much attention in the large-scale machine learning literature [9], [29], [45]. Even beyond these applications, we will show that the ℓ_∞ subspace sketch primitive in fact leads to the first $\mathrm{poly}(d,\log n)$ space, $\mathrm{poly}(d,\log n)$ distortion algorithm for a much wider variety of geometric problems, k-robust directional width, including ℓ_p subspace sketch for $p < \infty$, convex hull, Löwner–John ellipsoids, volume maximization, minimum-width spherical shell, and solving linear programs. Our results can thus be seen as a high-dimensional and high-distortion analogue of the fact that ε-kernels solve many streaming problems in the $(1+\varepsilon)$-distortion setting [2], [3].

Next, we study streaming subspace sketches. Here, we obtain a deterministic algorithm achieving $O(d^2\log n)$ bits of space and $O((d\log n)^{\frac{1}{2}-\frac{1}{p}})$ distortion, significantly improving upon the earlier deterministic one-pass algorithms of [26], which incurred a $\mathrm{poly}(n)$ factor in either the space complexity or distortion. This nearly matches the offline guarantee obtained by using Lewis weights [25], [41], [56], achieving optimal trade-offs.

Although our streaming ℓ_p subspace sketch achieves nearly optimal trade-offs, it is still possible to ask for improvements in these bounds, as well as faster algorithms, in the offline setting where we have unlimited access to $\mathbf{A}$. In a third contribution, in the offline setting, we construct ℓ_p subspace embeddings with nearly optimal trade-offs between space complexity and distortion, which shave all $\mathrm{poly}\log n$ factors off of the distortion. As a crucial step, we give a new elementary proof of a "change of density" theorem in geometric functional analysis due to Lewis and Tomczak-Jaegermann [42], by using Lewis weights [25], [41], [56]. This allows us to make the construction algorithmic, and in fact, nearly input sparsity time. Our space complexity upper bound matches a subspace sketch lower bound due to [43]. These subspace sketch lower bounds also witness the near tightness of our streaming ℓ_p subspace sketch algorithms. See Table II for a summary.

Furthermore, our fast algorithms for computing these ℓ_p subspace embeddings give the fastest known running times for ℓ_p regression and ℓ_p column subset selection,

TABLE I
RESULTS FOR ONE-PASS STREAMING ℓ_p SUBSPACE SKETCH IN THE "FOR ALL" MODEL. NEW RESULTS HIGHLIGHTED IN BLUE. WE SUPPRESS $O(\log n)$ FACTORS IN THE DISTORTION AND $\mathrm{poly}(\varepsilon^{-1},\log n)$ FACTORS IN SPACE. "OPTIMAL" MEANS THAT THE TRADE-OFF BETWEEN THE SPACE COMPLEXITY AND DISTORTION IS OPTIMAL, UP TO $\log n$ FACTORS (SEE TABLE II).

	Distortion	Space	Det.	Online	Optimal
$p=2$	1	d^2	✓		✓
$p>0$	$1+\varepsilon$	$d^{2\vee(\frac{p}{2}+1)}$			✓
$p\ge 1$	$d^{O(1/\gamma)}$	$n^\gamma d$	✓		
$p=\infty$	$\sqrt{d}$	d^2	✓	✓	✓
$p>2$	$d^{\frac{1}{2}-\frac{1}{p}}$	d^2	✓	✓	✓
$p>2$	$d^{\frac{1}{2}\left(1-\frac{q}{p}\right)}$	$d^{\frac{q}{2}+1}$			✓
$p=2$	$1+\varepsilon$	d^2	✓	✓	✓

TABLE II
OUR RESULTS FOR SUBSPACE SKETCH IN THE "FOR ALL" MODEL. RESULTS FOR THE "FOR EACH" MODEL REMOVE A FACTOR OF d FROM THE SPACE BOUNDS, FOR THE SAME DISTORTION. NEW RESULTS HIGHLIGHTED IN BLUE. WE SUPPRESS $\mathrm{poly}\log n$ FACTORS IN THE SPACE COMPLEXITY.

		Distortion	Space	
$p\in(0,2]$	UB	1	d^2	
$p>2$	UB	$d^{\frac{1}{2}-\frac{1}{p}}$	d^2	
$p>2$	UB	1	$d^{p/2+1}$	
$p>2y$	LB	$d^{\frac{1}{2}\left(1-\frac{q}{p}\right)}$	$d^{q/2+1}$	$2\le q\le p$
$p>2$	UB	$d^{\frac{1}{2}\left(1-\frac{q}{p}\right)}$	$d^{q/2+1}$	$2<q<p$
$p>0$	LB	$<\infty$	d^2	

[2]Here, $\mathbf{A}_i$ is the $i\times d$ matrix formed by the first i rows of $\mathbf{A}$.

TABLE III
RESULTS FOR FAST NUMERICAL LINEAR ALGEBRA IN ℓ_p FOR $p > 2$, WITH THE CURRENT MATRIX MULTIPLICATION TIME $\omega \approx 2.37286$ [6]. HERE, q IS ANY NUMBER BETWEEN 2 AND p. WE SUPPRESS $\mathrm{poly}(\log n, \varepsilon^{-1})$ FACTORS IN THE RUNNING TIME AND CONSTANT FACTORS IN THE DISTORTION. LR = ℓ_p LINEAR REGRESSION, CSS = k-COLUMN SUBSET SELECTION.

	Distortion	Time	
LR	$1+\varepsilon$	n^ω	[1]
	$1+\varepsilon$	$\mathsf{nnz}(\mathbf{A}) + d^{\frac{p}{2}\omega}$	[1], [25]
	$d^{\frac{1}{2}\left(1-\frac{q}{p}\right)}$	$\mathsf{nnz}(\mathbf{A}) + d^{\frac{q}{2}\omega}$	
CSS	$k^{1-\frac{1}{p}}$	$n^\omega d$	[1], [27]
	$k^{1-\frac{1}{p}}$	$\mathsf{nnz}(\mathbf{A})d + k^{\frac{p}{2}\omega} d$	[1], [25], [27]
	$k^{1-\frac{1}{p}+\frac{1}{2}\left(1-\frac{q}{p}\right)}$	$\mathsf{nnz}(\mathbf{A})d + k^{\frac{q}{2}\omega} d$	

when we allow for distortions which scale as $\mathrm{poly}(d)$ (see Table III). Note that algorithms for ℓ_p column subset selection already incur distortions on the order $\mathrm{poly}(d)$ [20], [27] (as they must due to known lower bounds).

B. Streaming Algorithms for Geometric Problems

We first introduce two models of streaming algorithms which we study: the *row arrival streaming model* and the *online coreset model*. In these models, we have an $n \times d$ input matrix $\mathbf{A}$ with rows $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n$, where n is so large that we cannot observe the entire matrix at once, and we can only observe one row at a time.

In the row arrival streaming model, we assume that $\mathbf{A} \in \mathbb{Z}^{n\times d}$ is an integer matrix with entries bounded by $\mathrm{poly}(n)$. Then, the rows $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n$ are presented in a stream one at a time in that order, and we must minimize the number of bits that we store while making only one pass[3] through the stream of rows of $\mathbf{A}$.

On the other hand, in the online coreset model, the input matrix $\mathbf{A}$ takes real values $\mathbb{R}^{n\times d}$. Again, the rows $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n$ are presented in one pass over a stream, one at a time, in that order. However, in this model, for each $i \in [n]$, we must irrevocably choose whether to store $\mathbf{a}_i$ or not. That is, if we choose to store $\mathbf{a}_i$, then we may not discard it at a later time. For each stored row, we allow for the row $\mathbf{a}_i$ to be scaled by some weight $\mathbf{w}_i \in \mathbb{R}$. The goal is to minimize the number of rows of $\mathbf{A}$ that are stored. We assume that we may perform exact arithmetic and linear algebra on the stored rows.

1) Online Coresets for ℓ_∞ Subspace Sketch: We first discuss our results for the ℓ_∞ subspace sketch problem, in both the row arrival streaming and online coreset models, which is the basis for all of our algorithms for geometric problems.

[3]We also consider algorithms which make multiple passes through the stream, but we will restrict to one pass for now.

Definition I.1 (Streaming/Online ℓ_∞ Subspace Sketch)**.** *The* streaming ℓ_∞ subspace sketch problem *is defined as follows*[4]*. We are given an $n \times d$ matrix $\mathbf{A}$ over one pass through a row arrival stream. Then:*

- *In the* row arrival streaming model*, $\mathbf{A} \in \mathbb{Z}^{n\times d}$ with entries bounded by* $\mathrm{poly}(n)$*, and we must maintain a data structure $Q : \mathbb{R}^d \to \mathbb{R}$ such that, at the end the stream, we have for some $\Delta \geq 1$ that*
$$\text{for all } \mathbf{x} \in \mathbb{R}^d, \qquad \|\mathbf{A}\mathbf{x}\|_\infty \leq Q(\mathbf{x}) \leq \Delta\|\mathbf{A}\mathbf{x}\|_\infty$$
- *In the* online coreset model*, $\mathbf{A} \in \mathbb{R}^{n\times d}$ is a real matrix, and we must irrevocably choose a subset of entries $S \subseteq [n]$ and weights $\mathbf{w} \in \mathbb{R}^S$ as well as a function $Q : \mathbb{R}^d \to \mathbb{R}$ depending only on* $\mathrm{diag}(\mathbf{w})\mathbf{A}|_S$ *such that, at the end of the stream, we have for some $\Delta \geq 1$ that*
$$\text{for all } \mathbf{x} \in \mathbb{R}^d, \qquad \|\mathbf{A}\mathbf{x}\|_\infty \leq Q(\mathbf{x}) \leq \Delta\|\mathbf{A}\mathbf{x}\|_\infty$$

To motivate and discuss the streaming ℓ_∞ subspace sketch problem, we first illustrate some connections with computational geometry. Note that Löwner–John ellipsoids can be used to achieve $\sqrt{d}$ distortion and d^2 words of space for ℓ_∞ subspace sketch in the offline setting, which is a nearly optimal trade-off. Thus, one may wonder whether there are algorithms for maintaining Löwner–John ellipsoids in a row arrival stream. This is, however, a fundamental unresolved problem in streaming computational geometry [5], [51]. In fact, we show that Löwner–John ellipsoids require $\Omega(n)$ bits of space to maintain up to a distortion of less than $\Theta(\sqrt{d/\log n})$:

Theorem I.2. *Any algorithm that maintains the Löwner–John ellipsoid of $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n \in \mathbb{R}^d$, up to a factor of $\sqrt{d/\log n}$, in one pass over a row arrival stream with probability at least $2/3$, must use $\Omega(n)$ bits space.*

This is perhaps surprising, given that for the syntactically similar MEB problem, $O(1)$ approximation is possible using $\mathrm{poly}(d, \log n)$ bits of space [5], [17]. Despite this, we obtain a deterministic streaming algorithm, and in fact an online coreset, for ℓ_∞ subspace embeddings:

Theorem I.3. *Let $\mathbf{A}$ be an $n \times d$ matrix presented in one pass over a row arrival stream. There is an algorithm $\mathcal{A}$ which maintains a coreset $S \subseteq [n]$ such that*
$$\text{for all } \mathbf{x} \in \mathbb{R}^d,\ \|\mathbf{A}|_S\mathbf{x}\|_\infty \leq \|\mathbf{A}\mathbf{x}\|_\infty \leq \Delta\|\mathbf{A}|_S\mathbf{x}\|_\infty$$
where

- *in the streaming model, $\Delta = O(\sqrt{d\log n})$, $|S| = O(d\log n)$, and $\mathcal{A}$ uses $O(d^2\log^2 n)$ bits of space.*
- *in the online coreset model, $\Delta = O(\sqrt{d\log(n\kappa^{\mathsf{OL}})})$ and $|S| = O(d\log(n\kappa^{\mathsf{OL}}))$.*

[4]Although one may define randomized versions of this problem [43], as we consider later, we restrict ourselves to deterministic algorithms in this section.

As we show, any data structure Q which satisfies

$$\mathbf{Pr}\{Q(\mathbf{x}) \leq \|\mathbf{A}\mathbf{x}\|_\infty \leq \Delta \cdot Q(\mathbf{x})\} \geq \frac{2}{3}$$

for each $\mathbf{x} \in \mathbb{R}^d$ must either have $\Delta = \Omega(\sqrt{d/\log n})$ or use $\Omega(n)$ bits of space. Furthermore, we show that if

$$\mathbf{Pr}\big\{\text{for all } \mathbf{x} \in \mathbb{R}^d,\ Q(\mathbf{x}) \leq \|\mathbf{A}\mathbf{x}\|_\infty \leq \Delta \cdot Q(\mathbf{x})\big\} \geq \frac{2}{3}$$

for any $\Delta < \infty$, then Q must use $\Omega(d^2)$ bits of space. Thus, our deterministic streaming algorithm achieves the best distortion and space that is possible for any randomized offline algorithm, up to $\operatorname{poly}\log n$ factors.

2) Techniques for Online ℓ_∞ Subspace Sketch:

a) Strawman Solutions: We first discuss certain natural coreset approaches to the streaming ℓ_∞ subspace sketch problem and why they do not work, in order to illustrate the difficulty of the problem. We assume for simplicity for now that all input vectors have norm $\Theta(1)$.

Intuitively, we want a small number of input rows that are well spread apart, so that we have a small number of rows that approximate the entire data set $\mathbf{A}$ in all directions. One way to do this is to add a new row to our coreset if and only if it has a small inner product, say at most some threshold $\tau = 1/\operatorname{poly}(d)$, with each of the stored rows. Certainly, such a row must be included in the coreset, otherwise that row itself as a query would fail to achieve a $1/\tau$-approximation. This can also be shown to yield a small coreset of size at most $\operatorname{poly}(d)$. However, such an algorithm could fail to store a row which is very well-aligned with an earlier row, but also has a tiny component pointing outside of the span of every other row, which means the coreset would fail to have any multiplicative error. One could try to fix this by adding the condition that we add a row if it increases the rank of the coreset; this also does not work, since there could be future rows which significantly increase the maximum component in this direction, but also have large inner product with the stored rows.

Another approach, which attempts to address the problem of having rows which increase the maximum component in a given direction, is to maintain the maximum component for $\operatorname{poly}(d)$ random directions. That is, one can first choose a set of $\operatorname{poly}(d)$ random directions S, and for each $\mathbf{v} \in S$, store the input row which has the maximum inner product with $\mathbf{v}$. However, it can be shown that $\operatorname{poly}(d)$ directions is in fact not enough to "catch" hidden growing components. Indeed, suppose that that input rows consist of the standard basis vectors $\pm\mathbf{e}_1$. These vectors will be stored. Then, suppose that the algorithm receives the vector $\mathbf{a} := (1-1/n)\mathbf{e}_1 + (1/n^{10})\mathbf{e}_2$. In order for this vector to be stored by a random vector $\mathbf{v}$, we must have that

$$\langle \mathbf{a}, \mathbf{v}\rangle = \big\langle (1-1/n)\mathbf{e}_1 + (1/n^{10})\mathbf{e}_2, \mathbf{v}\big\rangle > |\langle \mathbf{e}_1, \mathbf{v}\rangle|,$$

or $\mathbf{v}_2 \geq n^9|\mathbf{v}_1|$ by rearranging. The probability that this occurs for a random vector $\mathbf{v}$ is at most $O(1/n^9)$, and thus by a union bound over the $\operatorname{poly}(d)$ many random vectors, no direction stores $\mathbf{a}$. However, $\mathbf{a}$ has a component outside of the span of the previous rows, so even for vectors whose norms are within $1 \pm 1/\operatorname{poly}(n)$ factors of each other, this algorithm fails. It is easy to see that even if we store rows that increase the rank of the coreset, it would still fail to store rows which increase the component along $\mathbf{e}_2$ by $\operatorname{poly}(n)$ factors.

b) Our Approach.: We now give a high-level proof of our online ℓ_∞ coreset. We seek $S \subseteq [n]$ such that $\|\mathbf{A}\mathbf{x}\|_\infty \leq \Delta\|\mathbf{A}|_S\mathbf{x}\|_\infty$, so suppose we have maintained such an S, and let $\mathbf{a} \in \mathbb{R}^d$ be a new row. As hinted previously, we encounter a problem if there exists any direction $\mathbf{x} \in \mathbb{R}^d$ along which $\mathbf{a}$ updates the maximum component by more than a $\operatorname{poly}(d)$ factor. That is, if there exists $\mathbf{x} \in \mathbb{R}^d$ such that $|\langle \mathbf{a}, \mathbf{x}\rangle| \gg \|\mathbf{A}|_S\mathbf{x}\|_\infty$, then we must include $\mathbf{a}$ in our coreset. However, we are unable to analyze such an algorithm, due to the lack of structure of the ℓ_∞ norm. Now note that if $|S| = \operatorname{poly}(d, \log n)$, then $\|\mathbf{A}|_S\mathbf{x}\|_2 = \operatorname{poly}(d, \log n)\|\mathbf{A}|_S\mathbf{x}\|_\infty$, so using $\|\mathbf{A}|_S\mathbf{x}\|_2$ is just as good of a condition for adding $\mathbf{a}$. The advantage is that the ℓ_2 norm has much more structure than the ℓ_∞ norm, which we can use to bound the size of the coreset.

Suppose now that we add $\mathbf{a}$ to our coreset whenever there exists $\mathbf{x} \in \mathbb{R}^d$ such that $|\langle \mathbf{a}, \mathbf{x}\rangle|^2 \geq \|\mathbf{A}|_S\mathbf{x}\|_2^2$. In the language of numerical linear algebra, this corresponds to the condition that *the leverage score of* $\mathbf{a}$ *with respect to* $\mathbf{A}|_S$ *is at least* 1. With the connection to leverage scores, we are now in the position to bound the size of S. Note that in the final coreset $\mathbf{A}|_S$, we have by construction that every row $\mathbf{a}_i$ has leverage score at least 1 with respect to the previous rows. This can be phrased as the fact that all of the *online leverage scores* $\boldsymbol{\tau}_i^{\mathsf{OL}}$ of $\mathbf{A}|_S$ are at least 1. Now, it can be shown that the ith online leverage score bounds the incremental difference between the log-volume spanned by columns of the first i rows $\mathbf{A}_i$ of $\mathbf{A}$ and $\mathbf{A}_{i+1}$, which gives a bound of $O(d\log\kappa^{\mathsf{OL}})$ on the sum of online leverage scores, where $\kappa^{\mathsf{OL}} = \|\mathbf{A}\|_2 \max_{i=1}^n\|\mathbf{A}^-\|_2$ is the online pseudo condition number of $\mathbf{A}$ [13], [24]. This means that S must have at most $O(d\log\kappa^{\mathsf{OL}})$ rows. In turn, we can bound the distortion as

$$\begin{aligned}\|\mathbf{A}\mathbf{x}\|_\infty &= \max_{i=1}^n|\langle \mathbf{a}_i, \mathbf{x}\rangle| \leq \|\mathbf{A}|_S\mathbf{x}\|_2 \\ &\leq \sqrt{|S|}\|\mathbf{A}|_S\mathbf{x}\|_\infty \leq O(\sqrt{d\log\kappa^{\mathsf{OL}}})\|\mathbf{A}|_S\mathbf{x}\|_\infty.\end{aligned}$$

Although the κ^{OL} here is for the submatrix $\mathbf{A}|_S$, it can be shown that this is only a $\operatorname{poly}(n)$ factor away from κ^{OL} of $\mathbf{A}$. While this discussion contains a number of ideas for our online coreset algorithm for the ℓ_∞ subspace sketch problem, we still need to improve our result from $O(\sqrt{d\log\kappa^{\mathsf{OL}}})$ to $O(\sqrt{d\log n})$ distortion for integer matrices with entries bounded by $\mathrm{poly}(n)$ for the row arrival streaming model. For this, we will improve the bound on the sum of online leverage scores for such matrices. We discuss this result in the next section.

3) Techniques for Sharper Online Numerical Linear Algebra: We now discuss our techniques for improving the sum of online leverage scores for integer matrices with entries bounded by $\mathrm{poly}(n)$. Naïvely, the earlier condition number bound gives a bound of $O(d^2 \log n)$ by using that for such matrices, $\kappa \leq \mathrm{poly}(n)^d$ (see, e.g., [22, Lemma 4.1]). Note that κ can indeed be as large as $\exp(\mathrm{poly}(d))$, even for sign matrices [7]. We improve this to the following:

Theorem I.4 (Sum of Online Leverage Scores)**.** *Let* $\mathbf{A} \in \mathbb{Z}^{n\times d}$ *have entries bounded by* $\mathrm{poly}(n)$. *Then,* $\sum_{i=1}^n \boldsymbol{\tau}_i^{\mathsf{OL}}(\mathbf{A}) = O(d \log n)$.

We start with the proof of [24], which gives a bound of $O(d \log \kappa^{\mathsf{OL}})$. This is done by analyzing the quantity $\det(\mathbf{A}^\top \mathbf{A} + \lambda \mathbf{I}_d)$, for $\lambda = (\max_{i=1}^n \|\mathbf{A}_i^-\|_2)^{-1}$. This quantity is at most $O(\|\mathbf{A}\|_2)^d$, and can be shown to be lower bounded by $\exp(\frac{1}{2}\sum_i \boldsymbol{\tau}_i^{\mathsf{OL}}(\mathbf{A})) \cdot \det(\lambda \mathbf{I}_d)^d$ by the matrix determinant lemma, which gives $\det(\mathbf{A}_{i+1}^\top \mathbf{A}_{i+1} + \lambda \mathbf{I}_d) \geq \det(\mathbf{A}_i^\top \mathbf{A}_i + \lambda \mathbf{I}_d)\exp(\boldsymbol{\tau}_i^{\mathsf{OL}}(\mathbf{A})/2)$ where $\mathbf{A}_j$ is the first j rows of $\mathbf{A}$. Taking logarithms on both sides and rearranging yields that $\sum_{i=1}^n \boldsymbol{\tau}_i^{\mathsf{OL}}(\mathbf{A}) \leq O\left(d \log \frac{\|\mathbf{A}\|_2}{\lambda}\right) = O(d \log \kappa^{\mathsf{OL}})$. Now, one may question whether regularizing by λ is necessary, as it leads to the undesirable $\log \frac{1}{\lambda}$ factor. Indeed, we set $\lambda = 0$ and instead analyze the *pseudodeterminant* $\mathrm{pdet}(\mathbf{A}^\top \mathbf{A})$, which is the product of the nonzero eigenvalues. With this change, we have almost the same result, except that we must treat rows i which do not lie in $\mathrm{rowspan}(\mathbf{A}_{i-1})$ differently. In this case, $\mathrm{pdet}(\mathbf{A}_i^\top \mathbf{A}_i) \geq \mathrm{pdet}(\mathbf{A}_{i-1}^\top \mathbf{A}_{i-1})\|\mathbf{a}_i^\perp\|_2^2$ where $\mathbf{a}_i^\perp$ is the component of $\mathbf{a}_i$ orthogonal to $\mathrm{rowspan}(\mathbf{A}_{i-1})$. Now observe that the product of $\|\mathbf{a}_i^\perp\|_2^2$ for all rows i which do not lie in $\mathrm{rowspan}(\mathbf{A}_{i-1})$ is exactly the volume spanned by these vectors, which is a positive integer, and thus ≥ 1. We thus avoid the $\log \frac{1}{\lambda}$ factor and instead get the upper bound of $O(d \log n)$.

As a result of Theorem I.4, we immediately remove condition number dependencies from a variety of results in online numerical linear algebra which rely on Theorem I.4, and answer an open question of [13] on removing the condition number dependence from the online spectral approximation problem, under bit complexity assumptions.

Theorem I.5 (Online Coreset for Spectral Approximation)**.** *Let* $\mathbf{A} \in \mathbb{Z}^{n\times d}$ *have entries bounded by* $\mathrm{poly}(n)$. *There is a deterministic online coreset algorithm which outputs* $\tilde{\mathbf{A}}$ *such that* $(1-\varepsilon)\mathbf{A}^\top\mathbf{A} \preceq \tilde{\mathbf{A}}^\top \tilde{\mathbf{A}} \preceq (1+\varepsilon)\mathbf{A}^\top \mathbf{A}$ *and the number of rows in* $\tilde{\mathbf{A}}$ *is* $O(d(\log n)^2/\varepsilon^2)$.

We also implement the simpler sampling algorithm with a similar randomized guarantee.

Theorem I.6 (Online Coreset for Spectral Approximation via Leverage Score Sampling)**.** *Let* $\mathbf{A} \in \mathbb{Z}^{n\times d}$ *have entries bounded by* $\mathrm{poly}(n)$. *There is an online coreset algorithm which outputs* $\tilde{\mathbf{A}}$ *such that*

$$\mathbf{Pr}\left\{(1-\varepsilon)\mathbf{A}^\top\mathbf{A} \preceq \tilde{\mathbf{A}}^\top\tilde{\mathbf{A}} \preceq (1+\varepsilon)\mathbf{A}^\top\mathbf{A}\right\} \geq \frac{2}{3}$$

and the number of rows in $\tilde{\mathbf{A}}$ *is* $O(d(\log d)(\log n)/\varepsilon^2)$.

4) High-Dimensional Computational Geometry in Polynomial Space: We now show that our ℓ_∞ subspace sketch algorithm gives the first polynomial space algorithms for many important problems in streaming computational geometry, including the basic problems of symmetric width, convex hull, and Löwner–John ellipsoids. Previous algorithms for these problems had an exponential dependence on d, due to reliance on ε-kernels [2], [3]. In particular, in the high-dimensional regime of $d \geq C \log n$ for a large enough constant C, the memory bound for known results becomes larger than $\tilde{\Theta}(nd)$, and thus *there were no previously known nontrivial algorithms in this regime*, despite the fact that algorithms that work in the high-dimensional regime have been sought after for over a decade since they were suggested by [2], [3], [15], [61] and others.

In the following discussion, we assume a centrally symmetric instance, that is, if $\mathbf{a} \in \mathbb{R}^d$ is in the input point set, then so is $-\mathbf{a}$. Note that for most geometric problems falling under the class of *extent measure* problems [2], [5], considering only centrally symmetric instances is without loss of generality by translating to the origin, up to constant factor losses in the distortion.

Because our ℓ_∞ subspace sketch algorithm is online, many of our algorithms for streaming geometry are online as well, and we present results in both the row arrival streaming and online coreset models.

a) k-Robust Directional Width: Perhaps the most straightforward of our applications is directional width [2], [3], as this is equivalent to the ℓ_∞ subspace sketch problem. Using the "peeling" technique [4], we also obtain algorithms for *k-robust directional width* $\mathcal{E}_k(\mathbf{x}, \mathbf{A})$:

Theorem I.7 (k-Robust Directional Width)**.** *There is an algorithm* $\mathcal{A}$ *which maintains a coreset* $S \subseteq [n]$ *such that* $\frac{1}{\Delta}\mathcal{E}_k(\mathbf{x}, \mathbf{A}) \leq \mathcal{E}_k(\mathbf{x}, \mathbf{A}|_S) \leq \mathcal{E}_k(\mathbf{x}, \mathbf{A})$ *where*

- *in the streaming model,* $\Delta = O(\sqrt{d \log n})$, $|S| = O(kd \log n)$, *and* $\mathcal{A}$ *uses* $O(kd^2 \log^2 n)$ *bits of space.*
- *in the online coreset model,* $\Delta = O(\sqrt{d \log(n\kappa^{\mathsf{OL}})})$ *and* $|S| = O(kd \log(n\kappa^{\mathsf{OL}}))$.

b) Convex Hull: A fundamental problem in computational geometry is the approximation of the convex hull of n points $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n \in \mathbb{R}^d$. For $(1+\varepsilon)$-approximation, ε-kernels [2], [3] give coresets of near-optimal size of $\varepsilon^{-\Theta(d)}$, even in the streaming model [15], [16]. However, a general streaming algorithm for convex hull in $\mathrm{poly}(d, \log n)$ bits of space, even with $\mathrm{poly}(d, \log n)$ distortion, remained elusive. In the offline setting, this is possible via coresets for Löwner–John ellipsoids (see Section 3.6 of [57]).

By using our coreset for ℓ_∞ subspace sketch, we obtain coresets for approximating symmetric convex hulls, with $\mathrm{poly}(d, \log n)$ bits of space and distortion.

Theorem I.8 (Streaming Convex Hulls)**.** *There is an algorithm $\mathcal{A}$ which maintains a coreset $S \subseteq [n]$ such that $\mathrm{conv}(\{\pm\mathbf{a}_i\}_{i\in S}) \subseteq \mathrm{conv}(\{\pm\mathbf{a}_i\}_{i=1}^n) \subseteq \Delta\,\mathrm{conv}(\{\pm\mathbf{a}_i\}_{i\in S})$ where*

- *in the streaming model, $\Delta = O(\sqrt{d\log n})$, $|S| = O(d\log n)$, and $\mathcal{A}$ uses $O(d^2\log^2 n)$ bits of space.*
- *in the online coreset model, $\Delta = O(\sqrt{d\log(n\kappa^{\mathsf{OL}})})$ and $|S| = O(d\log(n\kappa^{\mathsf{OL}}))$.*

Note that this also gives us a $O(\sqrt{d\log n})^d$-factor approximation to the volume of convex hull.

c) Löwner–John Ellipsoids: As previously discussed, streaming Löwner–John ellipsoids in the high-dimensional setting has been open [5], [51]: [51] proposed a simple algorithm of iteratively adding points to a Löwner–John ellipsoid which does not yield $\mathrm{poly}(d, \log n)$ distortion, while [5] gave an $O(1)$-approximation for MEB in $\mathrm{poly}(d)$ space, and asked whether their ideas applied to Löwner–John ellipsoids.

We first note that our streaming ℓ_∞ subspace sketch result immediately gives a result for Löwner–John ellipsoids for linear inequality polytopes.

Theorem I.9 (Löwner–John Ellipsoids in Polynomial Space)**.** *Let $K = \{\mathbf{x}\in\mathbb{R}^d : \|\mathbf{A}\mathbf{x}\|_\infty \le 1\}$. There is an algorithm $\mathcal{A}$ which maintains a coreset $S \subseteq [n]$ from which we can compute an ellipsoid E' such that $E' \subseteq K \subseteq \Delta E'$ where*

- *in the streaming model, $\Delta = O(\sqrt{d\log n})$, $|S| = O(d\log n)$, and $\mathcal{A}$ uses $O(d^2\log^2 n)$ bits of space.*
- *in the online coreset model, $\Delta = O(\sqrt{d\log(n\kappa^{\mathsf{OL}})})$ and $|S| = O(d\log(n\kappa^{\mathsf{OL}}))$.*

Since $K \subseteq E \subseteq \sqrt{d}K$, E' is an $O(\Delta\sqrt{d})$-approximate Löwner–John ellipsoid.

We then show that taking polars yields Löwner–John ellipsoids for symmetric convex hulls as well.

d) Volume Maximization: We next consider the problem of selecting k rows that approximately maximizes the volume of the parallelepiped spanned by the rows, known as *volume maximization*, or maximum a posteriori (MAP) inference of determinantal point processes (DPPs) [10]. Relative error guarantees for this problem have been studied by [32], [33], [46], culminating in the following:

Theorem I.10 (Theorem 1.9 of [46])**.** *Let $C \in [1, (\log n)/k]$. There is a one-pass streaming algorithm that computes a subset $S \subseteq [n]$ of k points such that*

$$\mathbf{Pr}\Big\{O(Ck)^{k/2}\,\mathrm{Vol}(\mathbf{A}|_S) \ge \mathrm{Vol}(\mathbf{A}|_{S_*})\Big\} \ge \frac{2}{3}$$

where $\mathrm{Vol}(\mathbf{A}|_S)$ is the volume of the parallelepiped spanned by the rows $\mathbf{A}|_S$ indexed by S and $\mathbf{A}|_{S_*}$ is a set of k rows that maximizes the volume. The algorithm uses $O(n^{O(1/C)}d)$ bits of space.

This result is obtained by combining coresets for volume maximization [32] with streaming ε-kernels for directional width [15]. Note that even when $C = (\log n)/k$, the space complexity is $\exp(O(k))d$ and thus still exponential in k. By replacing ε-kernels for directional width with our ℓ_∞ subspace sketch result, we obtain the first relative error polynomial space algorithms for volume maximization[5].

Theorem I.11 (Streaming Volume Maximization)**.** *Let $1 < C < (\log n)/k$ and $r = (\log n)/C$. There is a one-pass streaming algorithm that computes a subset $S \subseteq [n]$ of k points such that*

$$\mathbf{Pr}\Big\{O(r^2Ck\log^2 n)^{k/2}\,\mathrm{Vol}(\mathbf{A}|_S) \ge \mathrm{Vol}(\mathbf{A}|_{S_*})\Big\} \ge \frac{2}{3}$$

where $\mathrm{Vol}(\mathbf{A}|_S)$ is the volume of the parallelepiped spanned by the rows $\mathbf{A}|_S$ indexed by S and $\mathbf{A}|_{S_*}$ is a set of k rows that maximizes the volume. The algorithm uses $O(rd\log^2 n)$ bits of space.

If only the indices (rather than the d-dimensional rows) are required, there is an algorithm using $O(k^2\log^3 n)$ bits of space with $O(k\log n)^k$ distortion.

e) Minimum-Width Spherical Shell: Our next application is the problem of approximating the *spherical shell* of minimum width which encloses a set of points. Formally, a spherical shell centered at $\mathbf{c} \in \mathbb{R}^d$ with inner radius r and outer radius R is $\sigma(\mathbf{c}, r, R) := \{\mathbf{x}\in\mathbb{R}^d : r \le \|\mathbf{x}-\mathbf{c}\|_2 \le R\}$, and we seek relative error approximations to $R - r$.

Theorem I.12 (Minimum Width Spherical Shell)**.** *Let $\mathbf{A}$ be an $n\times d$ matrix presented in one pass over a row arrival stream. There is an algorithm $\mathcal{A}$ which maintains a coreset $S \subseteq [n]$ from which we can compute find a center $\hat{\mathbf{c}}$, inner radius $\hat{r}$ and outer radius $\hat{R}$ such that $\sigma(\hat{\mathbf{c}}, \hat{r}, \hat{R}) \supseteq \{\mathbf{a}_i\}_{i=1}^n$ and $\hat{R} - \hat{r} \le \Delta^{3/2}\min_{\sigma(\mathbf{c},r,R)\supseteq\{\mathbf{a}_i\}_{i=1}^n} R - r$ where*

[5]The algorithm of [10] has polynomial space as well, but has an additive error guarantee

- *in the streaming model,* $\Delta = O(\sqrt{d \log n})$*,* $|S| = O(d \log n)$*, and* $\mathcal{A}$ *uses* $O(d^2 \log^2 n)$ *bits of space.*
- *in the online coreset model,* $\Delta = O(\sqrt{d \log(n\kappa^{\mathsf{OL}})})$ *and* $|S| = O(d \log(n\kappa^{\mathsf{OL}}))$*.*

f) Linear Programming.: Finally, we consider linear programming for instances with a centrally symmetric constraint polytope $\{\mathbf{x} \in \mathbb{R}^d : \|\mathbf{A}\mathbf{x}\|_\infty \leq 1\}$. More formally, we seek to approximate the optimal value of the following optimization problem

$$\begin{aligned} &\text{maximize} \quad \langle \mathbf{c}, \mathbf{x} \rangle \\ &\text{subject to} \quad \mathbf{x} \in \mathbb{R}^d, \|\mathbf{A}\mathbf{x}\|_\infty \leq 1 \end{aligned}$$

where the rows of $\mathbf{A}$ arrive in a row arrival stream.

Theorem I.13 (Streaming Linear Programming)**.** *Let* $\mathbf{A}$ *be an* $n \times d$ *matrix presented in one pass over a row arrival stream. Define the polytope* $K = \{\mathbf{x} \in \mathbb{R}^d : \|\mathbf{A}\mathbf{x}\|_\infty \leq 1\}$*. There is an algorithm* $\mathcal{A}$ *which maintains a coreset* $S \subseteq [n]$ *such that for any* $\mathbf{c} \in \mathbb{R}^d$*, one can compute from* $\mathbf{A}|_S$ *a vector* $\hat{\mathbf{x}} \in K$ *such that* $\max_{\mathbf{x} \in K} \langle \mathbf{c}, \mathbf{x} \rangle \leq \Delta \cdot \langle \mathbf{c}, \hat{\mathbf{x}} \rangle$ *where*

- *in the streaming model,* $\Delta = O(\sqrt{d \log n})$*,* $|S| = O(d \log n)$*, and* $\mathcal{A}$ *uses* $O(d^2 \log^2 n)$ *bits of space.*
- *in the online coreset model,* $\Delta = O(\sqrt{d \log(n\kappa^{\mathsf{OL}})})$ *and* $|S| = O(d \log(n\kappa^{\mathsf{OL}}))$*.*

C. Streaming and Online ℓ_p *Subspace Sketch*

1) The Subspace Sketch Problem: We now consider the ℓ_p subspace sketch problem, which is defined analogously to ℓ_∞ in Definition I.1. This problem in the offline setting, as well as its randomized variants, was introduced by [43]. When defining the randomized version of this guarantee, [43] define two versions, known as the "for each" and "for all" guarantees. For our streaming algorithms, we focus on the stronger "for all" guarantee.

Definition I.14. *Let* $\mathbf{A} \in \mathbb{R}^{n \times d}$ *and* $\Delta \geq 1$*. Then:*

- ***For each guarantee****:* Q_p *satisfies the* "for each" guarantee *if for each* $\mathbf{x} \in \mathbb{R}^d$*,*
$$\mathbf{Pr}\Big\{\|\mathbf{A}\mathbf{x}\|_p \leq Q_p(\mathbf{x}) \leq \Delta \|\mathbf{A}\mathbf{x}\|_p\Big\} \geq \frac{2}{3}$$
- ***For all guarantee****:* Q_p *satisfies the* "for all" guarantee *if*
$$\mathbf{Pr}\Big\{\forall \mathbf{x} \in \mathbb{R}^d, \|\mathbf{A}\mathbf{x}\|_p \leq Q_p(\mathbf{x}) \leq \Delta \|\mathbf{A}\mathbf{x}\|_p\Big\} \geq \frac{2}{3},$$

2) Prior Work on Streaming Subspace Sketch: The subspace sketch problem is a vast generalization of the more well-known *subspace embedding* problem, in which Q_p specifically takes the form $\|\mathbf{S}\mathbf{A}\mathbf{x}\|$, for some norm $\|\cdot\|$ and a linear map $\mathbf{S} \in \mathbb{R}^{s \times n}$. Many, but not all, of our upper bounds on the subspace sketch problem will actually be subspace embeddings.

In the regime of $\Delta = (1 + \varepsilon)$ for $\varepsilon \to 0$, near-optimal streaming algorithms can be obtained quite straightforwardly by leveraging ℓ_p subspace embeddings algorithms due to [25]. These subspace embedding results achieve near-optimal space complexity by sampling methods. One can then use the *merge-and-reduce* framework, in which one repeatedly finds subsets of rows that provide a $(1 + \varepsilon)$ approximation for blocks of rows, and then combines them in a binary tree fashion (see [13], [26]), to get streaming subspace embedding algorithms of approximately the same quality. Since the approximation is composed with a depth of $\log n$, our distortion is $(1 + \varepsilon)^{\log n}$; by replacing ε by $\frac{\varepsilon}{\log n}$, we recover the same trade-off as the offline subspace sketch problem, up to $\operatorname{poly} \log n$ factors. The space complexity is roughly $d^{2 \vee (p/2+1)}$ words of space. However, this is intractable when p is large.

The previous work of [26] studied the problem of maintaining a subspace sketch data structure *deterministically* using a similar merge-and-reduce strategy, but their results unfortunately incur an $n^{\Omega(1)}$ factor either in the distortion or the space complexity. Similar composable coreset approaches have been explored by other works, e.g., [32].

3) Streaming Algorithms for ℓ_p *Subspace Sketch:* We now discuss our results for the ℓ_p streaming subspace sketch problem. We first develop the following deterministic streaming algorithm, which greatly improves [26].

Theorem I.15. *Let* $\mathbf{A} \in \mathbb{Z}^{n \times d}$ *have entries bounded by* $\operatorname{poly}(n)$*. Let* $2 < p < \infty$*. There is a one-pass streaming algorithm maintaining a data structure* Q *using* $O(d^2 \log n)$ *bits of space such that for all* $\mathbf{x} \in \mathbb{R}^d$*,*

$$\|\mathbf{A}\mathbf{x}\|_p \leq Q(\mathbf{x}) \leq O((d \log n)^{\frac{1}{2} - \frac{1}{p}}) \|\mathbf{A}\mathbf{x}\|_p.$$

Our result proceeds by defining an online set of weights that behave similarly to Lewis weights.

By tolerating randomization and exponential time, we also obtain a full set of near-optimal trade-offs:

Theorem I.16. *Let* $\mathbf{A} \in \mathbb{Z}^{n \times d}$ *have entries bounded by* $\operatorname{poly}(n)$*. Let* $2 < q < p < \infty$*. There is a one-pass streaming algorithm which maintains a data structure* Q *using* $O(d^{q/2+1} \log n)$ *bits of space such that with probability at least* $2/3$*, for all* $\mathbf{x} \in \mathbb{R}^d$*,*

$$\|\mathbf{A}\mathbf{x}\|_p \leq Q(\mathbf{x}) \leq O(d^{\frac{1}{2}(1 - \frac{q}{p})} \log n) \|\mathbf{A}\mathbf{x}\|_p.$$

Furthermore, for $q = 2$, our result can be combined with ℓ_2 online coresets to yield online coresets for ℓ_p:

Corollary I.17. *Let* $2 < p < \infty$*. Let* $\mathbf{A}$ *be an* $n \times d$ *matrix presented in one pass over a row arrival stream. There is an algorithm* $\mathcal{A}$ *which maintains a coreset* $S \subseteq [n]$ *and weights* $\mathbf{w} \in \mathbb{R}^S$ *such that for all* $\mathbf{x} \in \mathbb{R}^d$*,*

$$\|\mathbf{A}\mathbf{x}\|_p \leq \|\operatorname{diag}(\mathbf{w}) \mathbf{A}|_S \mathbf{x}\|_2 \leq \Delta \|\mathbf{A}\mathbf{x}\|_p$$

where

- *in the streaming model, $\Delta = O((d\log n)^{\frac{1}{2}-\frac{1}{p}})$, $|S| = O(d\log n)$, and $\mathcal{A}$ uses $O(d^2\log^2 n)$ bits of space.*
- *in the online coreset model, $\Delta = O((d\log\kappa^{\mathsf{OL}})^{\frac{1}{2}-\frac{1}{p}})$ and $|S| = O(d\log\kappa^{\mathsf{OL}})$.*

As a corollary, we immediately obtain streaming algorithms for solving ℓ_p regression.

Corollary I.18 (Online Coresets for ℓ_p Regression)**.** *Let $2 < p \leq \infty$. Let $\mathbf{A}$ be an $n\times d$ matrix and let $\mathbf{b}$ be a vector and suppose that the $n\times(d+1)$ matrix $\mathbf{A}' \coloneqq [\mathbf{A}\ \mathbf{b}]$ is presented in a row arrival stream. There is an algorithm $\mathcal{A}$ which maintains a coreset $S\subseteq[n]$ and weights $\mathbf{w}\in\mathbb{R}^S$ from which we can compute $\hat{\mathbf{x}}\in\mathbb{R}^d$ such that $\|\mathbf{A}\hat{\mathbf{x}}-\mathbf{b}\|_p \leq \Delta\min_{\mathbf{x}}\|\mathbf{A}\mathbf{x}-\mathbf{b}\|_p$, where*

- *in the streaming model, $\Delta = O((d\log n)^{\frac{1}{2}-\frac{1}{p}})$, $|S| = O(d\log n)$, and $\mathcal{A}$ uses $O(d^2\log^2 n)$ bits of space.*
- *in the online coreset model, $\Delta = O((d\log\kappa^{\mathsf{OL}})^{\frac{1}{2}-\frac{1}{p}})$ and $|S| = O(d\log\kappa^{\mathsf{OL}})$.*

Our results are summarized in Table I.

D. Change of Density

We now turn to the offline ℓ_p subspace sketch problem. We first investigate *changes of density*:

Definition I.19 (Change of Density [36], [42])**.** *Let $0 < p,q \leq \infty$ and let $d\in\mathbb{N}$. Then, $c(d,p,q)$ denotes the smallest $c > 0$ such that for any $\mathbf{A}\in\mathbb{R}^{n\times d}$, there exists a nonnegative $\mathbf{w}\in\mathbb{R}^n$ such that, for $\mathbf{W} = \operatorname{diag}(\mathbf{w})$,*

$$\text{for all } \mathbf{x}\in\mathbb{R}^d, \qquad \|\mathbf{A}\mathbf{x}\|_p \leq \left\|\mathbf{W}^{\frac{1}{q}-\frac{1}{p}}\mathbf{A}\mathbf{x}\right\|_q \leq c\|\mathbf{A}\mathbf{x}\|_p.$$

Here, we think of $\mathbf{w}$ as weights (or a *measure*) on the rows of $\mathbf{A}$ when evaluating ℓ_q norms, i.e., $\|\mathbf{y}\|_{q,\mathbf{w}} = (\sum_{i=1}^n \mathbf{w}_i\cdot|\mathbf{y}_i|^q)^{1/q}$. Note then that $\left\|\mathbf{W}^{-1/p}\mathbf{A}\mathbf{x}\right\|_{p,\mathbf{w}} = \|\mathbf{A}\mathbf{x}\|_p$ so the map $\mathbf{A}\mathbf{x}\mapsto\mathbf{W}^{-1/p}\mathbf{A}\mathbf{x}$ equipped with the appropriate norm is an *isometry*. On the other hand, the weighted ℓ_q norm is $\left\|\mathbf{W}^{-1/p}\mathbf{A}\mathbf{x}\right\|_{q,\mathbf{w}} = \left\|\mathbf{W}^{1/q-1/p}\mathbf{A}\mathbf{x}\right\|_q$, which is comparable to $\|\mathbf{A}\mathbf{x}\|_p$ if $\mathbf{w}$ satisfies the guarantee of Definition I.19.

a) Lewis weights.: The following seminal result is known about the parameter $c(d,p,q)$ for $q = 2$:

Theorem I.20 ([35], [41], [56])**.** *Let $d\in\mathbb{N}$. For all $0 < p \leq \infty$, $c(d,p,2) = c(d,2,p) = d^{|1/2-1/p|}$.*

Theorem I.20 is due to Lewis [41] in the regime of $1 \leq p < \infty$, and the weights $\mathbf{w}$ achieving the guarantee of Definition I.19 are known as *Lewis weights*, in honor of [41]. For the remaining parameter regimes, the case of $p = \infty$ follows from Löwner–John ellipsoids [35], while the case of $0 < p < 1$ was proven in [56]. Although the original proof by Lewis in [41] uses involved theorems from Banach space theory, particularly the theory of p-summing operators [28], the proofs of [25], [56] notably provide elementary proofs based only on analyzing the Lagrange multipliers for a convex program.

The use of Lewis weights was introduced to the theoretical computer science community by [25], whose work made Lewis weights algorithmic by giving input sparsity time algorithms for approximating Lewis weights, and used them to obtain fast algorithms for solving ℓ_p regression. Subsequently, Lewis weights have become widely used in algorithms research, playing important roles in recent developments in optimization [30], [34], [40], convex geometry [37], randomized numerical linear algebra [19], [21], [25], [44], [47], and machine learning [18], [48], [52], [53]. Algorithms for computing Lewis weights themselves have also been refined over the years, both for $0 < p < \infty$ [31], [39], [40] as well as $p = \infty$, corresponding to Löwner–John ellipsoids [23], [57].

b) Change of density to ℓ_q, $q\neq 2$: The following gives an optimal bound on $c(d,p,q)$ for $q\neq 2$.

Theorem I.21 (Theorem 1.2 of [42])**.** *Let $d\in\mathbb{N}$ and let $1\leq p,q<\infty$. The following holds:*

- *If $\min(p,q)\leq 2$, then $c(d,p,q)\leq d^{\left|\frac{1}{q}-\frac{1}{p}\right|}$.*
- *If $\min(p,q)\geq 2$, then $c(d,p,q)\leq d^{\frac{1}{2}\left(1-\frac{p\wedge q}{p\vee q}\right)}$.*

As [42] show, the quantity $c(d,p,q)$ is intimately related to various other quantities, including p-summing norms and p-integral norms of operators, and is of independent interest in the functional analysis literature. For instance, an important corollary of this result is the best known upper bound on the *Banach–Mazur distance* [58] between a subspace of ℓ_p^n and any subspace of ℓ_q^n, which formalizes the notion of distance between ℓ_p and ℓ_q for subspaces. As the authors note in Corollary 1.9 [42], this result is optimal for $1\leq p<q<2$. In fact, we will show that the proof of a result of [43] implies that this is optimal in the regime of $\min(p,q)\geq 2$ as well, when n is large enough. Thus, Theorem I.21 obtains a tight characterization of the distance between subspaces of ℓ_p and ℓ_q, in the sense of Banach–Mazur distance.

For $\min(p,q)\leq 2$, Theorem I.21 follows from properties of Lewis weights, enjoying simple proofs and fast algorithms due to our refined understanding of Lewis weights. However, for $\min(p,q)\geq 2$, the proof is much more complicated. The authors first relate the problem of bounding $c(d,p,q)$ to bounding the smallest constant $\alpha > 0$ such that $\pi_q(u)\leq\alpha\pi_p(u)$ for all linear maps u (Definition 1.3 of [42])[6], where $\pi_p(u)$ is the *p-summing norm* of u [28]. To prove that α bounds $c(d,p,q)$, the

[6]In fact, they show that these two parameters are *equal*.

authors invoke a factorization theorem of Maurey [50][7], which replaces Lewis's theorem and gives weights $\mathbf{w}$ for the change of density. Finally, the bound on α follows from a result of [14], which uses results from the theory of operator ideals [54].

Our main result of this section is an elementary proof of Theorem I.21 using Lewis weights. Due to the simplicity of our proof, we obtain the following robust version of Theorem I.21, which refines [42] since:

1) The change of density is specifically the ℓ_p Lewis weights, rather than a tailor-made construction.
2) The error guarantees degrade gracefully when the change of density is replaced by an approximation.

Theorem I.22 (Change of density via approximate Lewis weights)**.** *Let* $\mathbf{A} \in \mathbb{R}^{n\times d}$ *and* $0 < p, q < \infty$. *Let* $\mathbf{w} \in \mathbb{R}^n$ *be* α*-approximate* ℓ_p *Lewis weight overestimates and* $\mathbf{W} = \operatorname{diag}(\mathbf{w})$. *There is* $\lambda_{d,p,q}$ *such that for all* $\mathbf{x} \in \mathbb{R}^d$,

$$\|\mathbf{A}\mathbf{x}\|_p \leq \left\|\lambda_{d,p,q} \cdot \mathbf{W}^{1/q-1/p}\mathbf{A}\mathbf{x}\right\|_q \leq \kappa_{d,p,q}\|\mathbf{A}\mathbf{x}\|_p$$

where

$$\kappa_{d,p,q} = \begin{cases} (\alpha d)^{\left|\frac{1}{q}-\frac{1}{p}\right|} & \text{if } \min(p,q) \leq 2 \\ (\alpha d)^{\frac{1}{2}\left(1-\frac{p\wedge q}{p\vee q}\right)} & \text{if } \min(p,q) \geq 2 \end{cases}$$

Our main technique is a new simple identity for Lewis weights which may be of independent interest, which shows that if we reweight the rows of $\mathbf{A}$ with ℓ_p Lewis weights, then the ℓ_q Lewis weights of the resulting matrix coincide with the ℓ_p Lewis weights of $\mathbf{A}$. Given this identity, the proof follows from just a few lines of estimates, which substantially simplifies the original proof of [42]. Furthermore, because our change of density uses Lewis weights, we inherit fast algorithms for computing these weights. Note that although polynomial time algorithms are known for many factorization theorems [59], known algorithms require solving constrained eigenvalue minimization problems, and are not known to have fast input sparsity time algorithms as Lewis weights do. Our result shows the following surprising message:

ℓ_p Lewis weights optimally approximate ℓ_p by ℓ_q.

We hope that our techniques will find further applications in functional analysis and theoretical computer science. In particular, we give the fastest known algorithms for ℓ_p linear regression with $\operatorname{poly}(d)$-factor relative error distortion and ℓ_p column subset selection. Both ℓ_p regression and ℓ_p column subset selection are extremely well-studied, and obtaining fast algorithms for these problems is important. See Table III for a summary.

[7] See also Proposition 10 in Chapter III.H of [60] for a proof and exposition in English of a similar theorem from [50], which gives the "transposed" result.

E. Subspace Sketch with Large Approximation

As an application of Theorem I.22, we obtain new tight bounds on the offline ℓ_p subspace sketch problem.

The offline subspace sketch problem captures the fundamental limits of dimension reduction in ℓ_p: with unbounded computation and access to $\mathbf{A}$, how much can $\mathbf{A}$ be compressed, as a function of the distortion Δ? The work of [43] studied this problem in the regime of $\Delta = (1+\varepsilon)$ for $\varepsilon \to 0$. Here, [43] found surprising separations between $p \in 2\mathbb{Z}$ and all other p, showing a lower bound of $\tilde{\Omega}(d/\varepsilon^2)$ bits of space required to store Q_p for $p \in [1,\infty) \setminus 2\mathbb{Z}$ for the "for each" guarantee, which separates these p from $p \in 2\mathbb{Z}$ due to an upper bound of $\tilde{O}(d^p)$ due to [55]. For $\varepsilon = \Theta(1)$, they showed a lower bound of $\tilde{\Omega}(d^{p/2})$ for the "for each" guarantee and $\tilde{\Omega}(d^{p/2+1})$ for the "for all" guarantee, matching known upper bounds.

Although space bounds of the form $\tilde{O}(d^{p/2}/\varepsilon^2)$ are possible for achieving $(1+\varepsilon)$ distortion [38], for large constant p, this space usage may already be problematic, especially if one is willing to tolerate a larger approximation factor. One could first observe that even for $p = \infty$, if one is willing to tolerate a distortion of $\sqrt{d}$, then it is possible to do better by using Löwner–John ellipsoids, since it only takes $O(d^2)$ words of space (or $O(d^2 \log n)$ bits) to store the quadratic form for the Löwner–John ellipsoid for the convex set $\{\mathbf{x} \in \mathbb{R}^d : \|\mathbf{A}\mathbf{x}\|_p \leq 1\}$. Taking this idea a step further, one could also store the quadratic form for the Lewis ellipsoid for $\mathbf{A}$ using $O(d^2)$ words to achieve a distortion of $O(d^{\frac{1}{2}-\frac{1}{p}})$. However, these two upper bounds jump from $d^{p/2+1}$ space to d^2 space, which raises the question of whether it is possible to obtain a smooth trade-off. As another contribution, we answer this question in the affirmative, by applying our Theorem I.22. Our trade-offs are summarized in Table II. The lower bounds of [43] extend to the parameter regime we consider, and shows that our upper bounds are nearly optimal, up to logarithmic factors. Our algorithmic technique is to first approximate the ℓ_p norm by the ℓ_q norm using Theorem I.22 with some $q < p$, and then to use a constant factor approximation to the ℓ_q norm using $O(d^{q/2})$ words of space for the "for each" guarantee or $\tilde{O}(d^{q/2+1})$ for the "for all" guarantee.

F. Independent and Concurrent Work

In [49] which appeared in COLT 2022, the authors give a one-pass streaming algorithm for approximating the Löwner–John ellipsoid of a convex hull which stores $O(d^2)$ floating point numbers and achieves a distortion of $O(\sqrt{d\log(R/r+1)})$, where R is the radius of the smallest ball containing the input points and r is the radius of the largest ball contained in the input points. The algorithm in [49] is quite different from ours, analyzing an algorithm similar to that of [51]. For real-valued inputs, their distortion is independent of n while our Theorem I.9 incurs a dependence on $\log n$. However, our algorithm offers a couple of other advantages over [49]: (1) for integer matrices with polynomially bounded entries, our result improves upon [49] by providing a $O(\sqrt{d \log n})$ distortion without further assumptions, whereas the aspect ratio R/r could be exponential in d; (2) our algorithm is a coreset algorithm, i.e., it only relies on storing a subset of the input points. We also note that they do not solve related ℓ_p subspace sketch algorithms, as we do.

Acknowledgment

We thank Timothy Chan, Sariel Har-Peled, Piotr Indyk, and Jeff Phillips for helpful feedback and suggestions. We thank Naren Manoj and Max Ovsiankin for pointing out an error in an earlier version of the draft. We thank anonymous reviewers for suggestions which helped improve the presentation of the draft.

References

[1] D. Adil, R. Kyng, R. Peng, and S. Sachdeva. Iterative refinement for ℓ_p-norm regression. In T. M. Chan, editor, *SODA 2019*, pages 1405–1424. SIAM, 2019.

[2] P. K. Agarwal, S. Har-Peled, and K. R. Varadarajan. Approximating extent measures of points. *J. ACM*, 51(4):606–635, 2004.

[3] P. K. Agarwal, S. Har-Peled, and K. R. Varadarajan. Geometric approximation via coresets. *Combinatorial and computational geometry*, 52(1-30):3, 2005.

[4] P. K. Agarwal, S. Har-Peled, and H. Yu. Robust shape fitting via peeling and grating coresets. *Discret. Comput. Geom.*, 39(1-3):38–58, 2008.

[5] P. K. Agarwal and R. Sharathkumar. Streaming algorithms for extent problems in high dimensions. *Algorithmica*, 72(1):83–98, 2015.

[6] J. Alman and V. V. Williams. A refined laser method and faster matrix multiplication. In D. Marx, editor, *SODA 2021*, pages 522–539. SIAM, 2021.

[7] N. Alon and V. H. Vũ. Anti-Hadamard matrices, coin weighing, threshold gates, and indecomposable hypergraphs. *J. Combin. Theory Ser. A*, 79(1):133–160, 1997.

[8] S. Arya, G. D. da Fonseca, and D. M. Mount. Optimal approximate polytope membership. In P. N. Klein, editor, *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017*, pages 270–288. SIAM, 2017.

[9] G. Ballard, T. G. Kolda, A. Pinar, and C. Seshadhri. Diamond sampling for approximate maximum all-pairs dot-product (MAD) search. In C. C. Aggarwal, Z. Zhou, A. Tuzhilin, H. Xiong, and X. Wu, editors, *2015 IEEE International Conference on Data Mining, ICDM 2015*, pages 11–20. IEEE Computer Society, 2015.

[10] A. Bhaskara, A. Karbasi, S. Lattanzi, and M. Zadimoghaddam. Online MAP inference of determinantal point processes. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020*, 2020.

[11] A. Bhaskara, S. Lattanzi, S. Vassilvitskii, and M. Zadimoghaddam. Residual based sampling for online low rank approximation. In D. Zuckerman, editor, *60th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2019*, pages 1596–1614. IEEE Computer Society, 2019.

[12] A. Blum, V. Braverman, A. Kumar, H. Lang, and L. F. Yang. Approximate convex hull of data streams. In I. Chatzigiannakis, C. Kaklamanis, D. Marx, and D. Sannella, editors, *45th International Colloquium on Automata, Languages, and Programming, ICALP 2018, July 9-13, 2018, Prague, Czech Republic*, volume 107 of *LIPIcs*, pages 21:1–21:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018.

[13] V. Braverman, P. Drineas, C. Musco, C. Musco, J. Upadhyay, D. P. Woodruff, and S. Zhou. Near optimal linear algebra in the online and sliding window models. In *FOCS 2020*, pages 517–528. IEEE, 2020.

[14] B. Carl. Inequalities between absolutely (p, q)-summing norms. *Studia Math.*, 69(2):143–148, 1980/81.

[15] T. M. Chan. Faster core-set constructions and data-stream algorithms in fixed dimensions. *Comput. Geom.*, 35(1-2):20–35, 2006.

[16] T. M. Chan. Dynamic streaming algorithms for epsilon-kernels. In S. P. Fekete and A. Lubiw, editors, *SoCG 2016*, volume 51 of *LIPIcs*, pages 27:1–27:11. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016.

[17] T. M. Chan and V. Pathak. Streaming and dynamic algorithms for minimum enclosing balls in high dimensions. *Comput. Geom.*, 47(2):240–247, 2014.

[18] X. Chen and M. Derezinski. Query complexity of least absolute deviation regression via robust uniform convergence. In M. Belkin and S. Kpotufe, editors, *COLT 2021*, volume 134 of *Proceedings of Machine Learning Research*, pages 1144–1179. PMLR, 2021.

[19] R. Chhaya, J. Choudhari, A. Dasgupta, and S. Shit. Streaming coresets for symmetric tensor factoriza-

tion. In *ICML 2020*, volume 119 of *Proceedings of Machine Learning Research*, pages 1855–1865. PMLR, 2020.

[20] F. Chierichetti, S. Gollapudi, R. Kumar, S. Lattanzi, R. Panigrahy, and D. P. Woodruff. Algorithms for ℓ_p low-rank approximation. In D. Precup and Y. W. Teh, editors, *ICML 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 806–814. PMLR, 2017.

[21] K. L. Clarkson, R. Wang, and D. P. Woodruff. Dimensionality reduction for tukey regression. In K. Chaudhuri and R. Salakhutdinov, editors, *ICML 2019*, volume 97 of *Proceedings of Machine Learning Research*, pages 1262–1271. PMLR, 2019.

[22] K. L. Clarkson and D. P. Woodruff. Numerical linear algebra in the streaming model. In M. Mitzenmacher, editor, *STOC 2009*, pages 205–214. ACM, 2009.

[23] M. B. Cohen, B. Cousins, Y. T. Lee, and X. Yang. A near-optimal algorithm for approximating the John ellipsoid. In A. Beygelzimer and D. Hsu, editors, *COLT 2019*, volume 99 of *Proceedings of Machine Learning Research*, pages 849–873. PMLR, 2019.

[24] M. B. Cohen, C. Musco, and J. Pachocki. Online row sampling. *Theory Comput.*, 16:1–25, 2020.

[25] M. B. Cohen and R. Peng. L_p row sampling by lewis weights. In R. A. Servedio and R. Rubinfeld, editors, *STOC 2015*, pages 183–192. ACM, 2015.

[26] G. Cormode, C. Dickens, and D. P. Woodruff. Leveraging well-conditioned bases: Streaming and distributed summaries in Minkowski p-norms. In J. G. Dy and A. Krause, editors, *ICML 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 1048–1056. PMLR, 2018.

[27] C. Dan, H. Wang, H. Zhang, Y. Zhou, and P. Ravikumar. Optimal analysis of subset-selection based l_p low-rank approximation. In H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, editors, *NeurIPS 2019*, pages 2537–2548, 2019.

[28] J. Diestel, H. Jarchow, and A. Tonge. *Absolutely summing operators*, volume 43 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, Cambridge, 1995.

[29] Q. Ding, H. Yu, and C. Hsieh. A fast sampling algorithm for maximum inner product search. In K. Chaudhuri and M. Sugiyama, editors, *The 22nd International Conference on Artificial Intelligence and Statistics, AISTATS 2019*, volume 89 of *Proceedings of Machine Learning Research*, pages 3004–3012. PMLR, 2019.

[30] D. Durfee, K. A. Lai, and S. Sawlani. ℓ_1 regression using lewis weights preconditioning and stochastic gradient descent. In S. Bubeck, V. Perchet, and P. Rigollet, editors, *COLT 2018*, volume 75 of *Proceedings of Machine Learning Research*, pages 1626–1656. PMLR, 2018.

[31] M. Fazel, Y. T. Lee, S. Padmanabhan, and A. Sidford. Computing lewis weights to high precision. In J. S. Naor and N. Buchbinder, editors, *SODA 2022*, pages 2723–2742. SIAM, 2022.

[32] P. Indyk, S. Mahabadi, S. O. Gharan, and A. Rezaei. Composable core-sets for determinant maximization: A simple near-optimal algorithm. In K. Chaudhuri and R. Salakhutdinov, editors, *ICML 2019*, volume 97 of *Proceedings of Machine Learning Research*, pages 4254–4263. PMLR, 2019.

[33] P. Indyk, S. Mahabadi, S. O. Gharan, and A. Rezaei. Composable core-sets for determinant maximization problems via spectral spanners. In S. Chawla, editor, *SODA 2020*, pages 1675–1694. SIAM, 2020.

[34] A. Jambulapati, Y. P. Liu, and A. Sidford. Improved iteration complexities for overconstrained *p*-norm regression. In S. Leonardi and A. Gupta, editors, *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing*, pages 529–542. ACM, 2022.

[35] F. John. Extremum problems with inequalities as subsidiary conditions. In *Studies and Essays Presented to R. Courant on his 60th Birthday, January 8, 1948*, pages 187–204. Interscience Publishers, Inc., New York, N. Y., 1948.

[36] W. B. Johnson and G. Schechtman. Finite dimensional subspaces of L_p. In *Handbook of the geometry of Banach spaces, Vol. I*, pages 837–870. North-Holland, Amsterdam, 2001.

[37] A. Laddha, Y. T. Lee, and S. S. Vempala. Strong self-concordance and sampling. In K. Makarychev, Y. Makarychev, M. Tulsiani, G. Kamath, and J. Chuzhoy, editors, *STOC 2020*, pages 1212–1222. ACM, 2020.

[38] M. Ledoux and M. Talagrand. *Probability in Banach spaces*. Classics in Mathematics. Springer-Verlag, Berlin, 2011.

[39] Y. T. Lee. *Faster algorithms for convex and combinatorial optimization*. PhD thesis, Massachusetts Institute of Technology, 2016.

[40] Y. T. Lee and A. Sidford. Solving linear programs with sqrt(rank) linear system solves. *CoRR*, abs/1910.08033, 2019.

[41] D. R. Lewis. Finite dimensional subspaces of L_p. *Studia Mathematica*, 63(2):207–212, 1978.

[42] D. R. Lewis and N. Tomczak-Jaegermann. Hilbertian and complemented finite-dimensional subspaces of Banach lattices and unitary ideals. *J. Functional Analysis*, 35(2):165–190, 1980.

[43] Y. Li, R. Wang, and D. P. Woodruff. Tight bounds

for the subspace sketch problem with applications. *SIAM J. Comput.*, 50(4):1287–1335, 2021.
[44] Y. Li, R. Wang, L. Yang, and H. Zhang. Nearly linear row sampling algorithm for quantile regression. In *ICML 2020*, volume 119 of *Proceedings of Machine Learning Research*, pages 5979–5989. PMLR, 2020.
[45] S. S. Lorenzen and N. Pham. Revisiting wedge sampling for budgeted maximum inner product search (extended abstract). In Z. Zhou, editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021*, pages 4789–4793. ijcai.org, 2021.
[46] S. Mahabadi, I. P. Razenshteyn, D. P. Woodruff, and S. Zhou. Non-adaptive adaptive sampling on turnstile streams. In K. Makarychev, Y. Makarychev, M. Tulsiani, G. Kamath, and J. Chuzhoy, editors, *STOC 2020*, pages 1251–1264. ACM, 2020.
[47] A. V. Mahankali and D. P. Woodruff. Optimal ℓ_1 column subset selection and a fast PTAS for low rank approximation. In D. Marx, editor, *SODA 2021*, pages 560–578. SIAM, 2021.
[48] T. Mai, C. Musco, and A. Rao. Coresets for classification - simplified and strengthened. In M. Ranzato, A. Beygelzimer, Y. N. Dauphin, P. Liang, and J. W. Vaughan, editors, *NeurIPS 2021*, pages 11643–11654, 2021.
[49] Y. Makarychev, N. S. Manoj, and M. Ovsiankin. Streaming algorithms for ellipsoidal approximation of convex polytopes. In P. Loh and M. Raginsky, editors, *Conference on Learning Theory*, volume 178 of *Proceedings of Machine Learning Research*, pages 3070–3093. PMLR, 2022.
[50] B. Maurey. *Théorèmes de factorisation pour les opérateurs linéaires à valeurs dans les espaces L^p*. Number 11 in Astérisque. Société Mathématique de France, Paris, 1974.
[51] A. Mukhopadhyay, A. Sarker, and T. Switzer. Approximate ellipsoid in the streaming model. In W. Wu and O. Daescu, editors, *COCOA 2010, Proceedings, Part II*, volume 6509 of *Lecture Notes in Computer Science*, pages 401–413. Springer, 2010.
[52] C. Musco, C. Musco, D. P. Woodruff, and T. Yasuda. Active sampling for linear regression beyond the ℓ_2 norm. *CoRR*, abs/2111.04888, 2021.
[53] A. Parulekar, A. Parulekar, and E. Price. L1 regression with Lewis weights subsampling. In *APPROX/RANDOM 2021*, volume 207 of *LIPIcs*, pages 49:1–49:21, 2021.
[54] A. Pietsch. *Operator ideals*, volume 20 of *North-Holland Mathematical Library*. North-Holland Publishing Co., Amsterdam-New York, 1980.
[55] G. Schechtman. Tight embedding of subspaces of L_p in ℓ_p^n for even p. *Proc. Amer. Math. Soc.*, 139(12):4419–4421, 2011.
[56] G. Schechtman and A. Zvavitch. Embedding subspaces of l_p into l_p^n, 0¡ p¡ 1. *Mathematische Nachrichten*, 227(1):133–142, 2001.
[57] M. J. Todd. *Minimum volume ellipsoids - theory and algorithms*, volume 23 of *MOS-SIAM Series on Optimization*. SIAM, 2016.
[58] N. Tomczak-Jaegermann. *Banach-Mazur distances and finite-dimensional operator ideals*, volume 38 of *Pitman Monographs and Surveys in Pure and Applied Mathematics*. Longman Scientific & Technical, Harlow; copublished in the United States with John Wiley & Sons, Inc., New York, 1989.
[59] J. A. Tropp. Column subset selection, matrix factorization, and eigenvalue optimization. In C. Mathieu, editor, *SODA 2009*, pages 978–986. SIAM, 2009.
[60] P. Wojtaszczyk. *Banach spaces for analysts*, volume 25 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, Cambridge, 1991.
[61] H. Zarrabi-Zadeh and T. M. Chan. A simple streaming algorithm for minimum enclosing balls. In *Proceedings of the 18th Annual Canadian Conference on Computational Geometry, CCCG 2006*, 2006.