Thesis Proposal:
Advances in Algorithms for Matrix Approximation via Sampling and Sketching

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

March 31, 2023

Thesis Committee

David P. Woodruff (Carnegie Mellon University, Chair)
Anupam Gupta (Carnegie Mellon University)
Richard Peng (Carnegie Mellon University)
Cameron Musco (University of Massachusetts Amherst)

Abstract

In the past two decades, the field of randomized numerical linear algebra has been extremely successful in developing algorithmic techniques for the problem of *matrix approximation*, in which large matrices are approximated by much smaller ones. This problem is fundamental to many areas of mathematics and computer science, and algorithmically, matrix approximation has found applications ranging from machine learning to graph algorithms to computational geometry and beyond.

In this thesis proposal, we further develop the theory of matrix approximation algorithms from the perspective of randomized numerical linear algebra, drawing particularly heavily from techniques based on sampling and sketching. We focus on obtaining *nearly optimal trade-offs* for fundamental problems in this literature, and succeed in resolving such bounds for problems including oblivious ℓ_p subspace embeddings, ℓ_p Lewis weight sampling, streaming Löwner–John ellipsoid approximation, active ℓ_p linear regression, and entrywise low rank approximation.

Contents

1 Introduction

Matrices are one of the most fundamental forms of representing data, and the problem of approximating large matrices by smaller or simpler matrices, or *matrix approximation*, is one of the most natural and classical problems in mathematics and computer science. As data-driven technologies proliferate throughout modern computer science, large matrices that represent enormous datasets have become some of the most central objects of study, and developing approximation algorithms for efficiently handling these matrices and datasets has become one of the most important computational challenges today. Furthermore, this problem has recently enjoyed a renewed focus in the past decade in the algorithms literature, thanks to the rise of the field of *randomized numerical linear algebra* [Mah11, Woo14, MT20]. Since its inception, the field has steadily matured and succeeded in mapping out the landscape of many of its cornerstone problems, elucidating the key gaps in our understanding that have yet to be settled. This thesis introduces and develops new techniques for randomized matrix approximation, with a focus on obtaining tight trade-offs for foundational problems in this literature.

1.1 Notation

Given positive real numbers $a > b > 0$ and $c > 0$, we let $x \in (a \pm b)c$ denote the statement that $(a-b)c \leq x \leq (a+b)c$. For a vector $\mathbf{y} \in \mathbb{R}^n$ and an index $i \in [n]$, we use the notation $\mathbf{y}_i$ and $\mathbf{y}(i)$ both to indicate the ith entry of $\mathbf{y}$. For a matrix $\mathbf{A} \in \mathbb{R}^{n \times d}$ we write $\mathbf{a}_i$ to denote the ith row of $\mathbf{A}$, $\mathbf{a}^j$ to denote the jth column of $\mathbf{A}$, and $\mathbf{A}_{i,j}$ for the (i,j)th entry of $\mathbf{A}$. We denote the Moore–Penrose pseudoinverse by $\mathbf{A}^-$.

Let $1 \leq p \leq \infty$. For a vector $\mathbf{x} \in \mathbb{R}^d$, we define the ℓ_p norm to be

$$\|\mathbf{x}\|_p = \begin{cases} \left(\sum_{i=1}^{d} |\mathbf{x}_i|^p \right)^{1/p} & p < \infty \\ \max_{i=1}^{d} |\mathbf{x}_i| & p = \infty \end{cases}$$

For a matrix $\mathbf{A} \in \mathbb{R}^{n \times d}$, we define the entrywise ℓ_p norm to be

$$\|\mathbf{A}\|_{p,p} = \begin{cases} \left(\sum_{i=1}^{n} \sum_{j=1}^{d} |\mathbf{A}_{i,j}|^p \right)^{1/p} & p < \infty \\ \max_{i=1}^{n} \max_{j=1}^{d} |\mathbf{A}_{i,j}| & p = \infty \end{cases}$$

1.2 Problems Studied

In this thesis proposal, we center many of our discussions around two problems that lie at the heart of matrix approximation, both in theory and in practice: *linear regression* and *low rank approximation*.

Subspace Embeddings and Linear Regression. In the linear regression problem, we have an input design matrix $\mathbf{A} \in \mathbb{R}^{n \times d}$ as well as a label vector $\mathbf{b} \in \mathbb{R}^n$, and we wish to find the "closest" linear combination $\mathbf{Ax}$ of the columns of $\mathbf{A}$ to the vector $\mathbf{b}$. The notion of "closest" is most often taken to be the ℓ_2 norm, also known as the least squares loss, and thus the problem we wish to solve is

$$\min_{\mathbf{x} \in \mathbb{R}^d} \|\mathbf{Ax} - \mathbf{b}\|_2^2.$$

This can be viewed as one of the simplest possible models for supervised learning, and is classically one of the most widely used prediction models. While far more complex models, especially those based on deep neural networks and other highly nonconvex models, are growing in popularity, linear regression is still a crucial building block for these more sophisticated algorithms. Furthermore, linear regression is the preferred choice as a predictive model itself in resource-limited settings and settings with high levels of noise or uncertainty.

As we will discuss further in this thesis proposal, the problem of linear regression is closely related to a notion of matrix approximation known as a *subspace embedding*, which, roughly speaking, is a way to approximate a matrix $\mathbf{A}$ and a vector $\mathbf{b}$ by another matrix $\mathbf{A}' = \mathbf{SA}$ and vector $\mathbf{b}' = \mathbf{Sb}$, such that solving linear regression using $\mathbf{A}'$ and $\mathbf{b}'$ is approximately as good as solving linear regression using $\mathbf{A}$ and $\mathbf{b}$. We will also show that subspace embeddings have applications beyond linear regression, by applying them to resolve an old problem in computational geometry (Section 2.2.5, [WY22a]).

We will discuss our contributions concerning subspace embeddings and linear regression in Section 2.

Low Rank Approximation. A related problem is *low rank approximation*, in which we wish to approximate an input matrix $\mathbf{A}$ by the "closest" rank k matrix $\mathbf{A}'$. Again, a popular choice for the formalization of "closest" is the least squares loss, also known as the *Frobenius norm*, and thus we wish to solve

$$\min_{\text{rank}(\mathbf{A}')\leq k} \|\mathbf{A} - \mathbf{A}'\|_F^2 = \min_{\mathbf{U}\in\mathbb{R}^{n\times k},\mathbf{V}\in\mathbb{R}^{k\times d}} \|\mathbf{A} - \mathbf{UV}\|_F^2.$$

Note that low rank approximation is also an important application of linear regression, due to the observation that low rank approximation can be solved by linear regression if one of the factors $\mathbf{U}$ or $\mathbf{V}$ are known. Thus, many algorithms for low rank approximation, including some that we study in this thesis proposal, are based on "guessing" one of $\mathbf{U}$ or $\mathbf{V}$ and then solving a linear regression problem.

Low rank approximation serves as one of the simplest models for unsupervised learning, and is widely used as a preprocessing step for "denoising" a dataset. Furthermore, when $\mathbf{U}$ is restricted to be formed from a small subset of the columns of $\mathbf{A}$, then low rank approximation problem serve as a method of feature selection [ABF+16].

We will discuss our contributions concerning low rank approximation in Section 3.

The Challenge of Robust and Sensitive Loss Functions. While our discussion so far has focused on the ℓ_2 norm as the loss function of choice, this may not always provide desired results in practice. For instance, consider the linear regression problem. We may view the least squares loss $\|\mathbf{Ax} - \mathbf{b}\|_2^2$ as an aggregation of the "fitting errors" $\langle \mathbf{a}_i, \mathbf{x}\rangle - \mathbf{b}_i$ on the n indices $i \in [n]$, in which we take the sum of the squared fitting errors. However, many other natural choices for ways to aggregate these fitting errors exist. For example, one could opt for a more "robust" or "average-case" notion of error by taking the loss to be the sum of the absolute values of the fitting errors, which gives the popular *least absolute deviations regression* problem, also known as the ℓ_1 linear regression problem, given by

$$\min_{\mathbf{x}\in\mathbb{R}^d} \|\mathbf{Ax} - \mathbf{b}\|_1.$$

Another extreme is to control the largest fitting error among the n indices $i \in [n]$, which gives a more "sensitive" or "worst-case" notion of error. This problem is known as the ℓ_∞ linear regression problem, and is given by

$$\min_{\mathbf{x}\in\mathbb{R}^d} \|\mathbf{Ax} - \mathbf{b}\|_\infty.$$

In general, one can in fact consider a smooth trade-off between these two extremes by considering the ℓ_p linear regression problem, given by

$$\min_{\mathbf{x}\in\mathbb{R}^d} \|\mathbf{Ax} - \mathbf{b}\|_p^p$$

for a parameter $1 \leq p \leq \infty$. While the flexibility of tuning the "sensitivity" of the loss function by varying the parameter p has proven to be extremely valuable in practice, handling these loss functions poses numerous additional challenges for developing algorithms. A large portion of this thesis proposal will be devoted to developing various techniques which will allow us to overcome these obstructions and extend many known results for the ℓ_2 loss to general ℓ_p losses, and in some cases to even more general loss functions such as the Huber loss and the Tukey loss (Section 2.2.6, [MMWY22]). We will also develop and apply related techniques for the low rank approximation problem, where we study a variety of different matrix losses including entrywise ℓ_p norms (Section 3.1.2, [WY23a]), general entrywise losses (Section 3.1.1, [WY23a]), cascading norms (Section 3.2, [WY23a]), and the spectral norm (Section 3.3, [WY22b]).

Streaming Models of Computation. A second focus of this thesis proposal is the development of algorithms for matrix approximation in *streaming models of computation*. In many practical settings of large-scale computation, it is impractical, or even impossible, to assume that the algorithm has access to the entire input, due to the sheer size of input instances. In such situations, it is more practical to assume that the algorithm interacts with its input by observing small pieces of the input at a time, which are fed into the algorithm sequentially. Such computational settings are well-modeled by *streaming algorithms*. A typical example is an algorithm which must solve linear regression (or any other empirical risk minimization problem) over an enormous dataset consisting of billions of training examples, where each individual training example is small enough to fit in memory, but the entire dataset consisting of all of the training examples will not. In this case, the typical approach is to sequentially read in the training examples one at a time from storage, while the algorithm updates its internal state in a small amount of memory. When our input matrix $\mathbf{A}$ arrives one row at a time, as in the previous example, we call this setting the *row arrival streaming* setting. We note that streaming models also are often useful for developing algorithms in *distributed models of computation* where the input dataset is sharded over multiple computers.

In addition to the row arrival model, we also study a related model of computation, known as the *online model*. This setting can be considered to be a restriction of the previous row arrival streaming setting, where as rows of the matrix $\mathbf{A}$ arrive, we need to make certain irrevocable decisions about the rows. In particular, we will study various forms of *online matrix approximation* problems, where our algorithm must output a subset of the rows which forms a good approximation to the input matrix $\mathbf{A}$, but the rows to be included in the subset must be selected irrevocably at each row arrival $i \in [n]$. The study of such online matrix approximation problems was introduced by [CMP16, CMP20], and we further develop the theory of online matrix approximation by studying the problems of ℓ_∞ subspace embeddings (Section 2.2.5, [WY22a]), ℓ_p subspace embeddings (Section 2.2.1, [WY23b]), and ℓ_p subspace approximation (Section 3.2, [WY23a]).

Finally, we consider a third form of streaming known as the *turnstile model of streaming*, in which our matrix $\mathbf{A}$ is presented as a stream of additive entrywise updates to the input matrix $\mathbf{A}$. That is, we receive entrywise updates of the form $\mathbf{A}_{i,j} \leftarrow \mathbf{A}_{i,j} + \Delta$ for some real number $\Delta \in \mathbb{R}$. This model of streaming significantly generalizes the row arrival model, and allows for entries of $\mathbf{A}$ to be updated and even deleted. Because the algorithm must support such general updates, the turnstile model of streaming severely restricts the form of the algorithm, and all known algorithmic approaches take the approach of maintaining *linear sketches*, in which the algorithm first chooses a linear map $\mathbf{S} \in \mathbb{R}^{m \times nd}$ and then maintains $\mathbf{S}\text{vec}(\mathbf{A})$, where $\text{vec}(\mathbf{A}) \in \mathbb{R}^{nd}$ is the nd-dimensional vector which represents the entries of $\mathbf{A}$. Note then that $\mathbf{S}\text{vec}(\mathbf{A})$ can be updated efficiently under the entrywise updates, using the linearity of $\mathbf{S}$. We will intensively study approaches to solving matrix approximation problems in the turnstile model of streaming in Section 2.1 [LWY21, WY23a].

2 Subspace Embeddings and Linear Regression

A *subspace embedding* is a notion of matrix approximation which considers an approximation $\mathbf{A}' = \mathbf{SA}$ to be close to a matrix $\mathbf{A}$ if the norms of vectors in the column space of $\mathbf{A}'$ are close to those of $\mathbf{A}$.

Definition 2.1 (Subspace embedding)**.** *Let $\mathbf{A} \in \mathbb{R}^{n \times d}$ and $\mathbf{S} \in \mathbb{R}^{r \times n}$. Let $\kappa \geq 1$ be a distortion parameter and let $\|\cdot\|$ be a norm. Then, $\mathbf{S}$ is a* κ-approximate subspace embedding *if for every $\mathbf{x} \in \mathbb{R}^d$,*

$$\|\mathbf{Ax}\| \leq \|\mathbf{SAx}\| \leq \kappa \|\mathbf{Ax}\|.$$

One of the most ideal settings for the application of subspace embeddings is for the design of efficient approximation algorithms for the least squares linear regression problem [DMM06a, Sar06]. Let $\mathbf{A} \in \mathbb{R}^{n \times d}$ be a design matrix and let $\mathbf{b} \in \mathbb{R}^n$ be a label vector, and suppose that we want to efficiently approximate

$$\min_{\mathbf{x} \in \mathbb{R}^d} \|\mathbf{Ax} - \mathbf{b}\|_2^2.$$

Furthermore, suppose that our matrix $\mathbf{A}$ is very tall, that is, $n \gg d$. Then, classically, this problem requires $O(nd^2)$ time to solve. Now suppose that we have an algorithm for efficiently computing a κ-approximate

subspace embedding $\mathbf{S} \in \mathbb{R}^{r \times n}$ in the ℓ_2 norm for the $n \times (d+1)$ matrix $[\mathbf{A}\ \mathbf{b}]$, that is, $\mathbf{A}$ together with $\mathbf{b}$ appended as an additional column. Note then that, for every $\mathbf{x} \in \mathbb{R}^d$,

$$\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 \leq \|\mathbf{S}\mathbf{A}\mathbf{x} - \mathbf{S}\mathbf{b}\|_2^2 \leq \kappa^2 \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2, \tag{1}$$

Now suppose we set

$$\hat{\mathbf{x}} \coloneqq \arg\min_{\mathbf{x} \in \mathbb{R}^d} \|\mathbf{S}\mathbf{A}\mathbf{x} - \mathbf{S}\mathbf{b}\|_2^2$$
$$\mathbf{x}^* \coloneqq \arg\min_{\mathbf{x} \in \mathbb{R}^d} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2$$

Then, $\hat{\mathbf{x}}$ is a κ^2-approximately optimal solution since

$$\begin{aligned}
\|\mathbf{A}\hat{\mathbf{x}} - \mathbf{b}\|_2^2 &\leq \|\mathbf{S}\mathbf{A}\hat{\mathbf{x}} - \mathbf{S}\mathbf{b}\|_2^2 && (1) \\
&\leq \|\mathbf{S}\mathbf{A}\mathbf{x}^* - \mathbf{S}\mathbf{b}\|_2^2 && \text{optimality of } \hat{\mathbf{x}} \\
&\leq \kappa^2 \|\mathbf{A}\mathbf{x}^* - \mathbf{b}\|_2^2 = \kappa^2 \min_{\mathbf{x} \in \mathbb{R}^d} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 && (1)
\end{aligned}$$

and furthermore, it can be computed in the time that it takes to compute $\mathbf{SA}$ and $\mathbf{Sb}$, plus only $O(rd^2)$ time. This can potentially be much faster than the original $O(nd^2)$ time, if $r \ll n$ and the computation of $\mathbf{SA}$ and $\mathbf{Sb}$ is fast. Indeed, this framework has been applied to develop some of the fastest known algorithms for least squares linear regression, as well as a variety of other related linear algebraic tasks [DMMW12, CW13, CCKW22, CSWZ23].

Algorithms for computing subspace embeddings generally fall under one of two classes: *oblivious* subspace embeddings and *non-oblivious* subspace embeddings, that is, those that depend on the input matrix $\mathbf{A}$ and those that do not. We obtain results in both classes, and discuss results for oblivious subspace embeddings in Section 2.1 and non-oblivious subspace embeddings in Section 2.2.

2.1 Oblivious Subspace Embeddings

Consider the problem of computing an ℓ_2 subspace embedding for $d = 1$, which simply corresponds to the problem of finding a norm-preserving linear map for a single vector. This is natural problem is resolved by a classical result due to Johnson and Lindenstrauss [JL84], which states that given a set $S \subseteq \mathbb{R}^n$ of m vectors in n dimensions, a random linear projection $\mathbf{S} \in \mathbb{R}^{r \times n}$ from n dimensions to $r = O(\varepsilon^{-2} \log m)$ dimensions has the property that

$$\|\mathbf{S}\mathbf{y}\|_2 = (1 \pm \varepsilon)\|\mathbf{y}\|_2$$

simultaneously for every $\mathbf{y} \in S$, with probability at least $2/3$. Thus, the ℓ_2 norm of a finite number of vectors can be preserved up to $(1 \pm \varepsilon)$ factors. Furthermore, the matrix $\mathbf{S}$, which is also known as a *sketch* in this context, can be taken to be *oblivious*, i.e., independent of the vectors to which it applies. Thus, for $d = 1$, oblivious subspace embeddings exist for the ℓ_2 norm with distortion $\kappa = (1 + \varepsilon)$. The fact that the sketch $\mathbf{S}$ can be taken to be oblivious allows it to be used in a wide variety of settings in which non-oblivious subspace embeddings may not apply, for example in streaming settings and distributed computation.

It may not be immediately clear that the technique of random projections also solves the problem of computing a subspace embedding for $d > 1$, since this involves preserving the ℓ_2 norm of every vector in the column space of $\mathbf{A}$, which is an *infinite* number of vectors, rather than a finite number m. Nevertheless, the following seminal result of Sarlos [Sar06] shows that random projections in fact do yield ℓ_2 subspace embeddings with distortion $(1 \pm \varepsilon)$.

Theorem 2.2 (Sarlos [Sar06])**.** *Let $\mathbf{S}$ be an $r \times n$ matrix of i.i.d. Gaussian random variables. There is an $r = O(\varepsilon^{-2} d \log d)$ such that for any $\mathbf{A} \in \mathbb{R}^{n \times d}$,*

$$\mathbf{Pr}\big\{\textit{for all } \mathbf{x} \in \mathbb{R}^d, \quad \|\mathbf{A}\mathbf{x}\|_2 \leq \|\mathbf{S}\mathbf{A}\mathbf{x}\|_2 \leq (1+\varepsilon)\|\mathbf{A}\mathbf{x}\|_2\big\} \geq \frac{99}{100}$$

that is, $\mathbf{S}$ is an ℓ_2 subspace embedding of $\mathbf{A}$ with distortion $(1+\varepsilon)$, with probability at least $99/100$.

Since the result of Theorem 2.2, a long line of work has studied further improvements to the development of oblivious ℓ_2 subspace embeddings [Sar06, CW13, NN13, Coh16, CCKW22, CSWZ23].

2.1.1 High Distortion Embeddings for ℓ_p [WY23a]

Given the result of Theorem 2.2, a natural question to ask is whether similar results exist for ℓ_p norms for $p \neq 2$. For $d = 1$, the question of whether the ℓ_p norm of a single vector can be preserved given linear measurements of the vector is well-studied in the *streaming* literature [SS02, BJKS04, IW05, Ind06], where for $p < 2$, $\tilde{\Theta}(\varepsilon^{-2})$ measurements is necessary and sufficient for approximation up to a factor of $(1+\varepsilon)$, while for $p > 2$, the ℓ_p norm cannot be approximated to within a constant factor unless $\Omega(n^{1-2/p})$ measurements are used. The latter result already prohibits a result of the form of Theorem 2.2 for $p > 2$, if the number of rows r of $\mathbf{S}$ must be subpolynomial in n. Thus, the key question is whether a theorem analogous to Theorem 2.2 is possible for $p < 2$.

One idea is to take inspiration from the proof of Theorem 2.2 as well as the classic streaming algorithm for ℓ_p norm estimation for vectors by [Ind06]. In Theorem 2.2, the sketch $\mathbf{S}$ can be taken to be a matrix with i.i.d. Gaussian entries [DG03], largely owing to the fact that the Gaussian distribution is 2-*stable*, that is, if $\mathbf{g} \in \mathbb{R}^n$ is an i.i.d. Gaussian vector and $\mathbf{y} \in \mathbb{R}^n$ is an arbitrary vector, then $\langle \mathbf{g}, \mathbf{y} \rangle$ is distributed as a single Gaussian random variable, scaled by $\|\mathbf{y}\|_2$. In fact, an analogous result is known for ℓ_p norms for $p < 2$:

Theorem 2.3 (Standard p-stable distributions [Ind06, Nol20])**.** *For $0 < p \leq 2$, there exists a probability distribution $\mathcal{D}_p$ called the* standard p-stable distribution *such that if $\mathbf{g} \in \mathbb{R}^n$ has entries drawn i.i.d. from $\mathcal{D}_p$, then for any $\mathbf{y} \in \mathbb{R}^n$ $\langle \mathbf{g}, \mathbf{y} \rangle$ is distributed as $\|\mathbf{y}\|_p g$, for $g \sim \mathcal{D}_p$.*

While Theorem 2.3 takes a step in the right direction, several challenges remain. For $p < 2$, the p-stable distributions $\mathcal{D}_p$ are *heavy-tailed* (unlike the 2-stable Gaussian distribution which enjoys sub-Gaussian tails), and in order to obtain $(1 \pm \varepsilon)$-approximate estimates with high probability, one usually needs to take a *median* of independent measurements of $|\langle \mathbf{g}, \mathbf{y} \rangle|$ to approximate $\|\mathbf{y}\|_p$. However, the approximation that we seek, of the form of Definition 2.1, would take a *mean* of the measurements, which in turn results in either a much higher distortion, or a much higher number of rows r for the sketch $\mathbf{S}$.

In fact, it turns out that this loss for $p < 2$ is inherent for oblivious ℓ_p subspace embeddings, as shown by [WW19, WW22]:

Theorem 2.4 (Lower bounds for oblivious ℓ_p subspace embeddings, [WW19, WW22])**.** *Suppose that a distribution $\mathcal{D}$ over $r \times n$ matrices $\mathbf{S}$ satisfies, for any $\mathbf{A} \in \mathbb{R}^{n \times d}$,*

$$\Pr_{\mathbf{S} \sim \mathcal{D}} \left\{ \text{for all } \mathbf{x} \in \mathbb{R}^d, \quad \|\mathbf{A}\mathbf{x}\|_p \leq \|\mathbf{S}\mathbf{A}\mathbf{x}\|_p \leq \kappa \|\mathbf{A}\mathbf{x}\|_p \right\} \geq \frac{99}{100}.$$

Then, the distortion κ is at least

$$\kappa = \Omega\left(\frac{1}{\frac{1}{d^{1/p}} \log^{2/p} r + \left(\frac{r}{n}\right)^{1/p - 1/2}} \right).$$

Note that typically, we seek to set $r = \text{poly}(d)$, which means that the distortion κ must be at least

$$\kappa = \Omega\left(\frac{d^{1/p}}{\log^{2/p} d} \right) = \tilde{\Omega}(d^{1/p})$$

and thus the distortion must be at least polynomial in d, and $(1+\varepsilon)$-approximations, or even $O(1)$-approximations, are not possible.

The first known upper bounds for ℓ_p subspace embeddings for $p < 2$ were obtained by [SW11], who obtained a construction for $r = \tilde{O}(d)$ rows and distortion $\kappa = \tilde{O}(d)$ for the case of $p = 1$. In fact, their sketch $\mathbf{S}$ is just constructed analogously to Theorem 2.2 with the 2-stable Gaussian distribution replaced by the 1-stable Cauchy distribution. That is, $\mathbf{S}$ is just an appropriate scaling of the $r \times n$ matrix where each entry is drawn independently from the standard 1-stable distribution, also known as the Cauchy distribution. Note that this result achieves a nearly optimal trade-off distortion for any $r = \text{poly}(d)$ rows by the lower bound of Theorem 2.4. While a dense Cauchy matrix is not as ideal to apply quickly, faster variants of this construction have been developed in subsequent works [MM13, WZ13, CDM+16, WW19, WW22].

With the trade-offs for oblivious ℓ_1 subspace embeddings being settled, the next natural question is to settle the analogous problem for $1 < p < 2$.

Question 2.5 ([WW19, WW22]). *Do there exist oblivious ℓ_p subspace embeddings that achieves the guarantee of Definition 2.1 for the ℓ_p norm with $\kappa = \tilde{O}(d^{1/p})$ and $r = \text{poly}(d)$?*

In fact, for a long time, the above question was thought to be resolved, and many papers claimed constructions of oblivious ℓ_p subspace embeddings achieving a distortion of $\kappa = \tilde{O}(d^{1/p})$ [MM13, WZ13, WW19]. Unfortunately, all of these results relied on the existence of a certain *well-conditioned basis*, whose proof contained an error, and the revised proofs only achieves constructions with a distortion of $\kappa = \tilde{O}(d)$ [WW22] for any $p \in (1, 2)$. Thus, the resolution of Question 2.5 became a central open question in the study of randomized matrix approximation [WW22].

In the work [WY23a], we give a positive resolution to Question 2.5:

Theorem 2.6 (Nearly optimal oblivious ℓ_p subspace embeddings [WY23a]). *Let $\mathbf{S}$ be an $r \times n$ matrix of i.i.d. p-stable random variables. There is an $r = \tilde{O}(d)$ such that for any $\mathbf{A} \in \mathbb{R}^{n\times d}$,*

$$\mathbf{Pr}\Big\{\text{for all } \mathbf{x} \in \mathbb{R}^d, \quad \|\mathbf{Ax}\|_p \le \|\mathbf{SAx}\|_p \le \tilde{O}(d^{1/p})\|\mathbf{Ax}\|_p\Big\} \ge \frac{99}{100}$$

that is, $\mathbf{S}$ is an ℓ_p subspace embedding of $\mathbf{A}$ with distortion $\kappa = \tilde{O}(d^{1/p})$, with probability at least 99/100.

As alluded to previously, our approach is to tackle the problem of the existence of a well-conditioned basis, which we discuss next. For the ℓ_2 norm, every subspace admits an *orthogonal basis*, which is a basis for the subspace which exactly preserves the ℓ_2 norm. That is, for any matrix $\mathbf{A} \in \mathbb{R}^{n\times d}$, there exists a matrix $\mathbf{U} \in \mathbb{R}^{n\times d}$ such that for any $\mathbf{x} \in \mathbb{R}^d$, there exists $\mathbf{x}' \in \mathbb{R}^d$ such that $\mathbf{Ax} = \mathbf{Ux}'$, and furthermore, $\|\mathbf{Ux}\|_2 = \|\mathbf{x}\|_2$ for every $\mathbf{x} \in \mathbb{R}^d$. The existence of orthogonal bases plays a key role in the analyses of oblivious ℓ_2 subspace embeddings. However, it is easy to see that for $p \neq 2$, exact analogues of orthogonal bases do not exist, and thus we must settle for an appropriately relaxed notion of "orthogonal bases". One way to meaningfully define such an analogue was introduced by [DDH+09], based on a similar definition by [Cla05]:

Definition 2.7 ((α, β, p)-well-conditioned basis, Definition 3, [DDH+09]). *Let $\mathbf{A} \in \mathbb{R}^{n\times d}$ be rank d matrix, let $p \ge 1$, and let $q = p/(p-1)$ be the Hölder dual of p. Then, $\mathbf{U} \in \mathbb{R}^{n\times d}$ is an (α, β, p)-well-conditioned basis if (1) $\|\mathbf{U}\|_{p,p} \le \alpha$ and (2) for any $\mathbf{z} \in \mathbb{R}^d$, $\|\mathbf{z}\|_q \le \beta\|\mathbf{Uz}\|_p$.*

Note that for ℓ_2, an orthogonal basis $\mathbf{U}$ corresponds to an $(\alpha, \beta, 2)$-well-conditioned basis with parameters $\alpha = d^{1/2}$ and $\beta = 1$. For ℓ_1, [SW11] showed that the well-known construction of Auerbach bases [Aue30] from the geometric functional analysis literature corresponds to an $(\alpha, \beta, 1)$-well-conditioned basis with $\alpha = d$ and $\beta = 1$. For $p \in (1, 2)$, however, the works of [MM13, WZ13, WW19], it was mistakenly claimed that Auerbach bases also give (α, β, p)-well-conditioned bases for $\alpha = d^{1/p}$ and $\beta = 1$, while they in fact only give $\alpha = d$ and $\beta = 1$.

While our techniques in [WY23a] do not give a construction for $(d^{1/p}, 1, p)$-well-conditioned basis, we in fact show that by relaxing the notion of well-conditioned bases to well-conditioned *spanning sets*, we can obtain a construction that is sufficient to prove Theorem 2.6. More specifically, we show the following:

Theorem 2.8 ((α, β, p)-well-conditioned spanning set, [WY23a]). *Let $\mathbf{A} \in \mathbb{R}^{n\times d}$, let $p \ge 1$, and let q be the Hölder dual of p. Then, there exists $\mathbf{U} \in \mathbb{R}^{n\times s}$ for $s = O(d \log\log d)$ such that (1) $\|\mathbf{U}\|_{p,p} \le s^{1/p}$ and (2) for any $\mathbf{x} \in \mathbb{R}^d$, there exists $\mathbf{z} \in \mathbb{R}^s$ such that $\mathbf{Ax} = \mathbf{Uz}$ and $\|\mathbf{z}\|_2 \le O(1)\|\mathbf{Uz}\|_p$.*

That is, we show that by relaxing the use of a basis, which is only allowed to contain d vectors, to a spanning set consisting of just $O(d \log\log d)$ vectors, we can obtain a spanning set which has properties which just as good as a $(\tilde{O}(d^{1/p}), O(1), p)$-well-conditioned basis (and in fact, better in some aspects). Our construction of the well-conditioned spanning set is in fact nothing more than a *coreset for John ellipsoids* [Tod16], which are a subset of $s = O(d \log\log d)$ vectors which approximate the minimum volume enclosing ellipsoid of a given set of vectors.

2.1.2 Low Distortion Embeddings for ℓ_1 [LWY21]

In Section 2.1.1, we studied algorithms for oblivious ℓ_p subspace embeddings with distortion κ on the order of $\text{poly}(d)$, as the lower bound of Theorem 2.4 prohibited a construction with smaller distortion, if

we insist on $r = \text{poly}(d)$. However, if we are allowed to make r as large as $\exp(\text{poly}(d))$, then the lower bound of Theorem 2.4 no longer gives a lower bound, and we can hope for a distortion of $\kappa = (1+\varepsilon)$. Indeed, [WW19, WW22] studied the question of whether $(1+\varepsilon)$ approximations are possible if we allow for superpolynomial dependencies on d, and showed that if $r = \exp(\exp(\text{poly}(d)))$, that is, doubly exponential in d, then a dense Cauchy embedding (similarly to that used in [SW11]) admits oblivious ℓ_1 subspace embeddings with $(1+\varepsilon)$ distortion. However, this leads to an *exponential* gap in the bound on r. A natural question is to resolve this gap:

Question 2.9 ([WW19, WW22])**.** *Do there exist oblivious ℓ_p subspace embeddings that achieves the guarantee of Definition 2.1 for the ℓ_p norm with $\kappa = (1+\varepsilon)$ and $r = \exp(\text{poly}(d, \varepsilon^{-1}))$?*

In [LWY21], we study Question 2.9 and answer it affirmatively for $p = 1$ with the following theorem:

Theorem 2.10 ($(1+\varepsilon)$ oblivious ℓ_1 subspace embeddings [LWY21])**.** *There exists a distribution over $r \times n$ matrices $\mathbf{S}$ for $r = \exp(\tilde{O}(d/\varepsilon))$ such that for any $\mathbf{A} \in \mathbb{R}^{n\times d}$,*

$$\mathbf{Pr}\big\{\textit{for all } \mathbf{x} \in \mathbb{R}^d, \quad \|\mathbf{Ax}\|_1 \le \|\mathbf{SAx}\|_1 \le (1+\varepsilon)\|\mathbf{Ax}\|_1\big\} \ge \frac{99}{100}$$

that is, $\mathbf{S}$ is an ℓ_1 subspace embedding of $\mathbf{A}$ with distortion $\kappa = (1+\varepsilon)$, with probability at least $99/100$.

Our sketch $\mathbf{S}$ for proving Theorem 2.10 requires a number of novel ideas. Instead of using the typical approach for oblivious ℓ_1 subspace embeddings based on Cauchy sketches [SW11, WW19, WW22], we instead start with the M-sketch of [CW15a], which is based on the classical techniques of hashing and subsampling from the streaming literature [IW05]. In the M-sketch, the sketching matrix $\mathbf{S}$ is taken to be a map which samples rows at a wide range of sampling probabilities, and then hashes the sampled rows into a smaller number of rows. In our application for $(1+\varepsilon)$ oblivious ℓ_1 subspace embeddings, it turns out that for any fixing of sampling probabilities p that we use, there is a hard instance vector such that its ℓ_1 mass is spread out over $\Theta(1/p)$ coordinates such that anti-concentration in the sampling process will cause our estimate of the ℓ_1 mass to be off by a $(1+\Omega(1))$ factor. To overcome this problem, we show how to *randomize the choice of the sampling probabilities themselves* so that we can avoid this worst-case scenario. While this idea is sufficient for preserving the ℓ_1 norm for one vector, we in fact need additional ideas in order to handle the entire subspace spanned by the columns of $\mathbf{A}$, which can increase r to be doubly exponential if done via a naive net argument. Instead, we show that in our setting, we can apply our single-vector analysis to the vector of ℓ_1 *sensitivities* (see Section 2.2.2), which implies norm preservation guarantee for the entire subspace via a novel argument.

Our techniques developed in this work have been further developed in [MOW23] to design streaming algorithms for logistic regression and ℓ_1 regression.

2.1.3 Future Directions for Oblivious ℓ_p Subspace Embeddings

While we have been able to resolve many of the outstanding gaps in our understanding of oblivious ℓ_p subspace embeddings, many interesting questions still remain to be explored. Perhaps one of the most notable unresolved challenges is to resolve the dependence on the accuracy parameter ε for $(1+\varepsilon)$ oblivious ℓ_p subspace embeddings. Our upper bounds in [LWY21] have a singly exponential dependence on $1/\varepsilon$, while there is no known lower bound which rules out an upper bound of the form $r = \exp(\text{poly}(d))/\text{poly}(\varepsilon)$. We conjecture that our upper bound is tight, and ask whether one can show an exponential lower bound in ε, even for $d = O(1)$.

Question 2.11. *Is there an $\exp(\text{poly}(1/\varepsilon))$ lower bound on r for $(1+\varepsilon)$ oblivious ℓ_p subspace embeddings for $d = O(1)$?*

2.2 Non-Oblivious Subspace Embeddings

In this section, we discuss our results on *non-oblivious* subspace embeddings. Unlike the results for oblivious subspace embeddings (Section 2.1), non-oblivious subspace embeddings can be constructed as a function of the input matrix $\mathbf{A}$, and thus admit subspace embeddings with much smaller distortion κ and number of rows r.

An important approach in the construction of non-oblivious subspace embeddings is *sampling*, where each of the r rows of the sketch $\mathbf{S}$ selects and scales just one row of $\mathbf{A}$. A prime example of this is the work of [DMM06a], which takes such a sampling approach to obtain one of the first randomized algorithms for linear regression via ℓ_2 subspace embeddings. In this work, the authors consider the *leverage scores* of the input matrix $\mathbf{A}$, which measures the importance of each of the n rows of $\mathbf{A}$.

Definition 2.12 (Leverage scores)**.** *Let $\mathbf{A} \in \mathbb{R}^{n\times d}$. Then for each $i \in [n]$, the i*th leverage score of $\mathbf{A}$ *is defined to be*

$$\boldsymbol{\tau}_i(\mathbf{A}) \coloneqq \sup_{\mathbf{A}\mathbf{x}\neq 0} \frac{[\mathbf{A}\mathbf{x}](i)^2}{\|\mathbf{A}\mathbf{x}\|_2^2} = \mathbf{a}_i^\top (\mathbf{A}^\top \mathbf{A})^- \mathbf{a}_i,$$

where $\mathbf{a}_i = \mathbf{e}_i^\top \mathbf{A}$ is the ith row of $\mathbf{A}$.

A series of works have culminated in the following guarantee for leverage score sampling.

Theorem 2.13 (Leverage score sampling [DMM06a, RV07])**.** *Let $\mathbf{A} \in \mathbb{R}^{n\times d}$. Let $\alpha > 0$ and let $p_i = \min\{1, \boldsymbol{\tau}_i(\mathbf{A})/\alpha\}$ for $i \in [n]$. Let $\mathbf{S} \in \mathbb{R}^{n\times n}$ be the diagonal matrix formed by independently setting*

$$\mathbf{S}_{i,i} = \begin{cases} \frac{1}{\sqrt{p}_i} & \text{with probability } p_i \\ 0 & \text{with probability } 1 - p_i \end{cases}$$

for each $i \in [n]$. Then, with probability at least 99/100, there is an α such that $\mathbf{S}$ is an ℓ_2 subspace embedding satisfying Definition 2.1 with $\kappa = (1+\varepsilon)$, and furthermore, $\mathbf{S}$ has at most $r = O(\varepsilon^{-2} d \log d)$ nonzero rows.

Algorithms for ℓ_2 subspace embeddings using leverage score sampling have subsequently been improved in [SS11, DMMW12, LMP13, CLM+15].

2.2.1 Lewis Weight Sampling and ℓ_p Subspace Embeddings [WY23b]

Given the result of Theorem 2.13, the next natural question, in a similar line of inquiry as Section 2.1.1, is whether analogous results can be obtained for the ℓ_p norm or not.

Sampling algorithms for ℓ_p subspace embeddings. One possible generalization of leverage scores to the ℓ_p setting comes from the observation that the leverage scores can be characterized as the row norms of any orthogonal basis of $\mathbf{A}$. That is, if $\mathbf{U} \in \mathbb{R}^{n\times d}$ is an orthogonal basis of $\mathbf{A} \in \mathbb{R}^{n\times d}$, then it is not hard to see that

$$\boldsymbol{\tau}_i(\mathbf{A}) = \left\|\mathbf{e}_i^\top \mathbf{U}\right\|_2^2$$

for every $i \in [n]$. We can then recall constructions of well-conditioned bases $\mathbf{U}$ for subspaces of ℓ_p (Definition 2.7) and define analogous scores that are proportional to $\left\|\mathbf{e}_i^\top \mathbf{U}\right\|_p^p$. Indeed, such approaches were considered and used to obtain ℓ_p subspace embeddings with $r = \text{poly}(d/\varepsilon)$ rows and $\kappa = (1+\varepsilon)$ distortion [DDH+09]:

Theorem 2.14 (ℓ_p leverage score sampling [Cla05, DDH+09])**.** *Let $\mathbf{A} \in \mathbb{R}^{n\times d}$ and $1 \le p < \infty$. Let $\mathbf{U} \in \mathbb{R}^{n\times d}$ be a $(\text{poly}(d), 1, p)$-well-conditioned basis for the column space of $\mathbf{A}$ (see Definition 2.7). Let $\alpha > 0$ and let $p_i = \min\{1, \|\mathbf{e}_i^\top \mathbf{U}\|_p^p/\alpha\}$ for $i \in [n]$. Let $\mathbf{S} \in \mathbb{R}^{n\times n}$ be the diagonal matrix formed by independently setting*

$$\mathbf{S}_{i,i} = \begin{cases} \frac{1}{p_i^{1/p}} & \text{with probability } p_i \\ 0 & \text{with probability } 1 - p_i \end{cases}$$

for each $i \in [n]$. Then, with probability at least 99/100, there is an α such that $\mathbf{S}$ is an ℓ_p subspace embedding satisfying Definition 2.1 with $\kappa = (1+\varepsilon)$, and furthermore, $\mathbf{S}$ has at most $r = \text{poly}(d/\varepsilon)$ nonzero rows.

Note that this result already separates oblivious ℓ_p subspace embeddings from non-oblivious ℓ_p subspace embeddings for $p \neq 2$ due to the lower bound of Theorem 2.4, which is perhaps surprising given that this separation does not exist for $p = 2$, where oblivious subspace embeddings can match the row and distortion trade-off of the best non-oblivious subspace embeddings. In fact, it turns out that one can hope for even better than Theorem 2.14, using the technique of *Lewis weight sampling*.

Lewis weight sampling. The work of [CP15] observed that the problem of constructing ℓ_p subspace embeddings of the form of Definition 2.1 has actually been studied decades ago in the geometric functional analysis literature, and obtains *nearly optimal* trade-offs between the number of rows r and the accuracy parameter ε. Indeed, a series of works [Lew78, BLM89, LT91, SZ01] culminated in the following result:

Theorem 2.15 (ℓ_p subspace embeddings, existential version [Lew78, BLM89, LT91, SZ01])**.** *Let $\mathbf{A} \in \mathbb{R}^{n\times d}$ and $0 < p < \infty$. Then, there exists an ℓ_p subspace embedding $\mathbf{S} \in \mathbb{R}^{r\times n}$ with distortion $\kappa = (1+\varepsilon)$ with*

$$r = \begin{cases} O(\varepsilon^{-2} d (\log d)^2 \log(d/\varepsilon)) & 0 < p < 2 \\ O(\varepsilon^{-2} d^{p/2} (\log d)^2 \log(d/\varepsilon)) & 2 < p < \infty \end{cases}$$

We note that the statement of Theorem 2.15 is slightly suboptimal in the logarithmic factors compared to the best known results [Tal90, Tal95, Zva00], but we present this version as it uses a simpler proof that we work extensively with, while achieving the best known dependencies on d and ε, up to polylogarithmic factors.

It has recently been shown that the upper bound of Theorem 2.15 is nearly optimal for $p < 2$, while for $p > 2$, the dependence on ε and d are individually optimal [LWW21, LLW23] when $d = \Omega(\log(1/\varepsilon))$. In fact, the lower bound of [LWW21] applies to *any* data structures which has the same guarantee as an ℓ_p subspace embedding:

Theorem 2.16. *Let $p \in [1,\infty) \setminus 2\mathbb{Z}$. Suppose that $\mathcal{A}$ is any randomized algorithm which processes any matrix $\mathbf{A} \in \mathbb{R}^{n\times d}$ into a data structure $\mathcal{Q}$ which supports queries $\mathbf{x} \in \mathbb{R}^d$ and outputs an estimate $\mathcal{Q}(\mathbf{x})$ such that*

$$\|\mathbf{A}\mathbf{x}\|_p \le \mathcal{Q}(\mathbf{x}) \le (1+\varepsilon)\|\mathbf{A}\mathbf{x}\|_p.$$

Then, $\mathcal{Q}$ requires $\tilde{\Omega}(d^2/\varepsilon^2)$ bits of space. Furthermore, for $p > 2$, $\mathcal{Q}$ requires $\tilde{\Omega}(\varepsilon^{-1} d^{p/2})$ bits of space.

The proof of Theorem 2.15 is *almost* algorithmic, as the proof is based on the probabilistic method; the only component which is not algorithmic is the construction of a certain set of weights known as the *Lewis weights* [Lew78], which can be viewed as a certain generalization of the leverage scores (Definition 2.12) for ℓ_p that differs from the ℓ_p leverage scores considered by [Cla05, DDH+09] in Theorem 2.14.

Definition 2.17 (ℓ_p Lewis weights [Lew78, CP15])**.** *Let $\mathbf{A} \in \mathbb{R}^{n\times d}$ and $0 < p < \infty$. Then, the ℓ_p* Lewis weights *of $\mathbf{A}$ are the unique set of weights $\mathbf{w} \in \mathbb{R}^n_{>0}$ such that for every $i \in [n]$,*

$$\mathbf{w}_i = \boldsymbol{\tau}_i(\mathbf{W}^{1/2-1/p}\mathbf{A}),$$

where $\mathbf{W} = \mathrm{diag}(\mathbf{w})$. We will denote the ℓ_p Lewis weights of $\mathbf{A}$ as $\mathbf{w}_i^p(\mathbf{A})$ for $i \in [n]$.

The work of Cohen and Peng [CP15] addresses the problem of the *algorithmic computation* of Lewis weights by showing that Lewis weights can, in fact, be approximated efficiently, and even in nearly input sparsity time for $p \in (0,4)$. Follow-up works have further refined algorithms for approximating Lewis weights [Lee16, CCLY19, FLPS22, JLS22], and Lewis weights can now be approximated in nearly input sparsity time for all $p > 0$ [JLS22].

While the works above address the question of approximating Lewis weights, using the Lewis weights to sample ℓ_p subspace embeddings is an orthogonal direction of investigation. By an appropriate adaptation of the earlier work in geometric functional analysis [BLM89, LT91, SZ01], as well as the construction of ℓ_p Lewis weights due to [CP15], one can obtain algorithmic constructions of ℓ_p subspace embeddings which match the guarantees of Theorem 2.15 [MMWY22]. However, this construction has the drawback that the sampling algorithm requires a sophisticated *recursive* structure in which the number of rows are reduced by half for $O(\log n)$ recursive rounds of sampling. This hinders the use of Lewis weight sampling in one-pass streaming settings [WY23b], and demonstrates a gap from algorithms for ℓ_2 leverage score sampling, which admits ℓ_2 subspace embeddings just by sampling proportionally to the leverage scores in a "one-shot" sampling algorithm [DMM06a, RV07], as well as streaming variants [CMP16, CMP20]. Indeed, the work of [CP15] studies the problem of obtaining ℓ_p subspace embeddings via sampling algorithms that simply sample rows proportionally to the Lewis weights in a "one-shot" manner analogous to leverage score sampling as in Theorem 2.13, rather than using a recursive sampling algorithm. In fact, such results are possible, and [CP15] obtain the following result:

Theorem 2.18 (ℓ_p Lewis weight sampling [CP15])**.** *Let $\mathbf{A} \in \mathbb{R}^{n\times d}$ and $1 \le p < \infty$. Let $\alpha > 0$ and let $p_i = \min\{1, \mathbf{w}_i^p(\mathbf{A})/\alpha\}$ for $i \in [n]$. Let $\mathbf{S} \in \mathbb{R}^{n\times n}$ be the diagonal matrix formed by independently setting*

$$\mathbf{S}_{i,i} = \begin{cases} \frac{1}{p_i^{1/p}} & \text{with probability } p_i \\ 0 & \text{with probability } 1 - p_i \end{cases}$$

for each $i \in [n]$. Then, with probability at least $99/100$, there is an α such that $\mathbf{S}$ is an ℓ_p subspace embedding satisfying Definition 2.1 with $\kappa = (1+\varepsilon)$, and furthermore, $\mathbf{S}$ has at most r nonzero rows, for

$$r = \begin{cases} O(\varepsilon^{-2} d \log(d/\varepsilon)) & p = 1 \\ O(\varepsilon^{-2} d \log(d/\varepsilon) \log\log(d/\varepsilon)) & 1 < p < 2 \\ O(\varepsilon^{-5} d^{p/2} (\log d) \log(1/\varepsilon)) & 2 < p < \infty \end{cases}$$

However, a notable gap exists between the algorithmic results of Theorem 2.18 based on "one-shot" sampling versus the existential results of Theorem 2.15 for $p > 2$ and its algorithmic version based on recursive sampling, where Theorem 2.15 achieves a quadratic dependence on ε, while Theorem 2.18 incurs a dependence of ε^5. An important question in the study of ℓ_p Lewis weight sampling is whether this gap can be closed:

Question 2.19. *For $p > 2$, can the guarantee of one-shot ℓ_p Lewis weight sampling in Theorem 2.18 be improved to $\tilde{O}(\varepsilon^{-2} d^{p/2})$?*

One of the main results we obtain in [WY23b] is a positive resolution to Question 2.19:

Theorem 2.20 (ℓ_p Lewis weight sampling, improved [WY23b])**.** *Let $\mathbf{A} \in \mathbb{R}^{n\times d}$ and $2 < p < \infty$. Then, Theorem 2.18 holds with*

$$r = O(\varepsilon^{-2} d^{p/2} (\log d)^2 \log(d/\varepsilon)).$$

In the work of [CP15], one of the major obstructions towards achieving a result like Theorem 2.20 is the lack of important structural properties of Lewis weights which hold for $p \le 2$ but not for $p > 2$. In particular, for $p \le 2$ ℓ_p Lewis weights satisfy *monotonicity* with respect to row additions:

Lemma 2.21 (Monotonicity of ℓ_p Lewis Weights, Lemma 5.5, [CP15])**.** *Let $\mathbf{A} \in \mathbb{R}^{n\times d}$ and $0 < p \le 2$. Let $m \ge n$ and let $\mathbf{A}' \in \mathbb{R}^{m\times d}$ be a matrix such that $\mathbf{e}_i^\top \mathbf{A} = \mathbf{e}_i^\top \mathbf{A}'$ for all $i \in [n]$, that is, $\mathbf{A}'$ is obtained by adding rows to $\mathbf{A}$. Then,*

$$\mathbf{w}_i^p(\mathbf{A}) \ge \mathbf{w}_i^p(\mathbf{A}')$$

for every $i \in [n]$.

This property is crucial in a reduction argument used by [CP15], which reduces the problem of proving guarantees for one-shot Lewis weight sampling to the problem of proving guarantees for uniform sampling of a different matrix. However, the monotonicity property of Lemma 2.21 fails to hold for $p > 2$, which obstructs the use of this reduction. Instead, the result for $p > 2$ in Theorem 2.18 is obtained by a chaining argument due to [BLM89], which directly analyzes the one-shot Lewis weight sampling algorithm but has a looser dependence on ε.

In [WY23b], we show how to directly circumvent the issue of non-monotonicity in the reduction argument used by [CP15]. The starting point of our idea is to first observe that in many cases, Lewis weights can be replaced by a substantially weakened version of Lewis weights that only satisfies a "one-sided" guarantee, which were introduced in [JLS22, WY22a]:

Definition 2.22 (One-sided ℓ_p Lewis weights)**.** *Let $\mathbf{A} \in \mathbb{R}^{n\times d}$ and $0 < p < \infty$. Let $\alpha > 0$. Then, $\mathbf{w} \in \mathbb{R}^n_{\ge 0}$ are α-approximate one-sided ℓ_p Lewis weights if*

$$\mathbf{w}_i \ge \alpha \cdot \boldsymbol{\tau}_i(\mathbf{W}^{1/2-1/p}\mathbf{A}),$$

where $\mathbf{W} = \mathrm{diag}(\mathbf{w})$.

We show in [WY23b] that relaxed ℓ_p Lewis weights of the form of Definition 2.22 still satisfy similar sampling guarantees as those studied [BLM89, LT91, SZ01]. It turns out that when we allow for a relaxation of the ℓ_p Lewis weights as in Definition 2.22, then we can define a version of ℓ_p Lewis weights *with respect to another matrix*, which is analogous to a notion studied for leverage scores in [CLM+15].

Lemma 2.23 (Lemma 4.6, [WY23b]). *Let $\mathbf{A} \in \mathbb{R}^{n \times d}$ and $2 \leq p < \infty$. Let $\mathbf{M} \in \mathbb{R}^{d \times d}$ be a symmetric positive semidefinite matrix. Then, there exist weights $\mathbf{w} \in \mathbb{R}^n_{\geq 0}$ such that for $i \in [n]$,*

$$\mathbf{w}_i = \left(\frac{p}{2}\right)^{\frac{p/2}{1-2/p}} \left(\mathbf{a}_i^\top (\mathbf{A}^\top \mathbf{W}^{1-2/p} \mathbf{A} + \mathbf{M})^- \mathbf{a}_i\right)^{2/p}$$

and

$$\sum_{i=1}^n \mathbf{w}_i \leq \left(\frac{p}{2}\right)^{\frac{1}{1-2/p}} d.$$

Lemma 2.23 can also be viewed as a version of *batch online* Lewis weights, where we initially have a matrix $\mathbf{B}$ as well as Lewis weights for them, and then receive a new batch of rows $\mathbf{A}$ which we must append, and still define a notion of ℓ_p Lewis weights that is "consistent" for the entire matrix obtained by concatenating $\mathbf{B}$ with $\mathbf{A}$. This perspective reveals how Lemma 2.23 is useful for proving Theorem 2.20: it allows us to circumvent non-monotonicity of exact ℓ_p Lewis weights, and "extend" ℓ_p Lewis weights of a matrix after a batch of row additions! Indeed, this is the central idea of the proof of Theorem 2.20.

Our ideas here also lead to the first *online* ℓ_p subspace embeddings which achieve guarantees which nearly match those of Theorems 2.18 and 2.20, in the setting where the rows $\mathbf{a}_i$ of $\mathbf{A}$ must be sampled as they arrive one by one in a stream, and cannot be accessed again if a row is not chosen to be sampled. This provides a generalization of the results of [CMP16, CMP20] for ℓ_2 subspace embeddings to ℓ_p subspace embeddings, and answers open questions of [BDM+20] and [CLS22].

2.2.2 Sensitivity Sampling [WY23c]

So far, we have discussed two possible generalizations of leverage score sampling: one based on using well-conditioned bases (Theorem 2.14), and one based on Lewis weights (Theorems 2.18 and 2.20). In fact, there is another natural candidate, known as ℓ_p *sensitivities*, which we study in this section.

The *sensitivity sampling framework* was introduced by [LS10, FL11] and further optimized by [BFL16, FSS20] in order to develop a unified approach to sampling-based approximation algorithms for a wide range of problems including clustering, projective clustering, low rank approximation and subspace approximation, empirical risk minimization, and others. In this general framework, we seek to approximate an objective function $f : X \to \mathbb{R}_{\geq 0}$ of the form of a sum

$$f(\mathbf{x}) \coloneqq \sum_{i=1}^n f_i(\mathbf{x})$$

by sampling a subset $S \subseteq [n]$ as well as associated weights $\mathbf{w}_i$ for $i \in S$, so that

$$f(\mathbf{x}) = (1 \pm \varepsilon) \sum_{i \in S} \mathbf{w}_i f(\mathbf{x}) \tag{2}$$

simultaneously for every $\mathbf{x} \in X$. Note that if we set $X = \mathbb{R}^d$ and $f_i(\mathbf{x}) = |\langle \mathbf{a}_i, \mathbf{x} \rangle|^p$ for each $i \in [n]$, then this corresponds to the problem of sampling an ℓ_p subspace embedding. To sample our approximation, we consider the *sensitivity scores* $\boldsymbol{\sigma}_i$, and sample functions f_i with probabilities p_i proportional to $\boldsymbol{\sigma}_i$ with weights $\mathbf{w}_i = 1/p_i$.

Definition 2.24 (Sensitivity score [LS10, FL11]). *For $i \in [n]$, let $f_i : X \to \mathbb{R}_{\geq 0}$ be functions. Then, the ith* sensitivity score *is defined as*

$$\boldsymbol{\sigma}_i \coloneqq \sup_{\mathbf{x} \in X} \frac{f_i(\mathbf{x})}{\sum_{j=1}^n f_j(\mathbf{x})}$$

and the total sensitivity *is defined as $\mathfrak{S} \coloneqq \sum_{i=1}^n \boldsymbol{\sigma}_i$.*

In a wide variety of applications, it can be shown that sampling $r = \tilde{O}(\varepsilon^{-2}\mathfrak{S}d)$ functions f_i is sufficient to achieve the guarantee of (2), where d is the VC-dimension of a certain set system associated with the functions $\{f_i\}_{i=1}^n$ [LS10, FL11, BFL16, FSS20].

In this section, we study sensitivity sampling when specialized to the seting of ℓ_p subspace embeddings, and introduce the following definition:

Definition 2.25 (ℓ_p sensitivities)**.** *Let $\mathbf{A} \in \mathbb{R}^{n\times d}$ and $p \geq 1$. Then, for each $i \in [n]$, we define the i*th ℓ_p sensitivity *to be*

$$\boldsymbol{\sigma}_i^p(\mathbf{A}) \coloneqq \sup_{\mathbf{x}\in\mathbb{R}^d, \mathbf{A}\mathbf{x}\neq 0} \frac{|[\mathbf{A}\mathbf{x}](i)|^p}{\|\mathbf{A}\mathbf{x}\|_p^p}$$

and the total ℓ_p sensitivity *to be $\mathfrak{S}^p(\mathbf{A}) \coloneqq \sum_{i=1}^n \boldsymbol{\sigma}_i^p(\mathbf{A})$.*

In this setting, the VC-dimension parameter d from the sensitivity sampling framework is equal to the dimension d of subspace spanned by the columns of $\mathbf{A}$, up to constant factors. Furthermore, it can be shown that the total ℓ_p sensitivity $\mathfrak{S}^p(\mathbf{A})$ is at most d for $p < 2$ and at most $d^{p/2}$ for $p > 2$ [WY22a]. Thus, sensitivity sampling immediately applies in this setting, and has been indeed used in the past to obtain ℓ_p subspace embeddings in settings Lewis weight sampling may not immediately apply [BDM+20, BHM+21].

Remark 2.26. *The calculation of ℓ_p sensitivities can be formulated as an ℓ_p regression problem, and can be computed efficiently using recent developments in algorithms for ℓ_p regression. Indeed, it is easy to see that*

$$\frac{1}{\boldsymbol{\sigma}_i^p(\mathbf{A})} = \min_{[\mathbf{A}\mathbf{x}](i)=1} \|\mathbf{A}\mathbf{x}\|_p^p,$$

which can be efficiently approximated to high precision in nearly matrix multiplication time [AKPS19, APS19, AS20].

Note that for $p = 2$, the ℓ_p sensitivities are exactly equal to the leverage scores (Definition 2.12). This means that the ℓ_2 sensitivities always have a total sensitivity of $\mathfrak{S}^2(\mathbf{A}) = d$, and the general sensitivity sampling bound of $\tilde{O}(\varepsilon^{-2}\mathfrak{S}d) = \tilde{O}(\varepsilon^{-2}d^2)$ is quadratically worse than the nearly optimal guarantee of leverage score sampling of Theorem 2.13. However, for $p > 2$, ℓ_p sensitivity sampling in fact has the potential to produce a *smaller* number of rows than ℓ_p Lewis weight sampling, if the total sensitivity $\mathfrak{S}^p(\mathbf{A})$ is small. Indeed, $\mathfrak{S}^p(\mathbf{A})$ can be as small as d even for $p > 2$, in which case one can obtain a sample complexity of $\tilde{O}(\varepsilon^{-2}\mathfrak{S}d) = \tilde{O}(\varepsilon^{-2}d^2)$ for such matrices, while Lewis weight sampling would require $\tilde{O}(\varepsilon^{-2}d^{p/2})$, which is much worse for $p > 4$. Thus, despite the fact that Lewis weight sampling already achieves nearly optimal bounds in the worst case (see Section 2.2.1), the study of sensitivity sampling using the scores of Definition 2.25 is still interesting for two reasons:

1. The definition of sensitivities can be massively generalized to a wide variety of sampling-based approximation problems.
2. For $p > 2$, sensitivity sampling admits matrix-dependent bounds which can circumvent the lower bounds of Theorem 2.16.

For these reasons, our work in [WY23c] studies the problem of obtaining the tightest possible bounds for ℓ_p sensitivity sampling:

Question 2.27. *What is the smallest sample complexity possible for the ℓ_p sensitivity sampling algorithm?*

While we are not able to complete resolve Question 2.27, we make progress towards this by giving an analysis of ℓ_p sensitivity sampling which goes beyond the general case bound of $\tilde{O}(\varepsilon^{-2}\mathfrak{S}d)$:

Theorem 2.28 (ℓ_p sensitivity sampling [WY23c])**.** *Let $1 \leq p < \infty$ and let $\mathbf{A} \in \mathbb{R}^{n\times d}$. Let $\alpha > 0$ and let $p_i = \min\{1, 1/n + \boldsymbol{\sigma}_i^p(\mathbf{A})/\alpha\}$ for $i \in [n]$. Let $\mathbf{S} \in \mathbb{R}^{n\times n}$ be the diagonal matrix formed by independently setting*

$$\mathbf{S}_{i,i} = \begin{cases} \dfrac{1}{p_i^{1/p}} & \text{with probability } p_i \\ 0 & \text{with probability } 1 - p_i \end{cases}$$

for each $i \in [n]$. Then, with probability at least 99/100, there is an α such that $\mathbf{S}$ is an ℓ_p subspace embedding satisfying Definition 2.1 with $\kappa = (1+\varepsilon)$, and furthermore, $\mathbf{S}$ has at most r nonzero rows, for

$$r = \begin{cases} \varepsilon^{-2}\mathfrak{S}^p(\mathbf{A})^{2/p} \operatorname{poly}\log n & 1 \le p < 2 \\ \varepsilon^{-2}\mathfrak{S}^p(\mathbf{A})^{2-2/p} \operatorname{poly}\log n & 2 < p < \infty \end{cases}$$

In fact, our improved analysis of ℓ_p sensitivity sampling is largely based off of the analysis of ℓ_p Lewis weight sampling in the works of [BLM89, LT91]. One of the key aspects of the analysis of Lewis weight sampling in these works is the use of a sophisticated *chaining argument* to replace a simpler net argument. When Lewis weights are used as sampling probabilities, then such a chaining argument goes through due to the fact that the resulting matrix has uniformly bounded leverage scores, which in turn is a consequence of the specific definition of Lewis weights. However, when we instead use the ℓ_p sensitivities, we no longer have this property, and the analysis needs to be modified.

To address this problem, we observe that although ℓ_p sensitivity sampling does not directly lead to uniformly bounded leverage scores, it *does* lead to uniformly bounded ℓ_p *sensitivities* in the resulting matrix. We then show that this in turn implies approximately uniformly bounded leverage scores, by relating the ℓ_p sensitivities to the leverage scores. Upon making this observation, Theorem 2.28 follows by relatively natural modifications to the arguments of [BLM89, LT91] as well as [CP15].

In fact, our techniques and observations used to obtain Theorem 2.28 lead to improved guarantees for yet another generalization of ℓ_2 leverage score sampling, known as *root leverage score sampling*. In root leverage score sampling, the sampling probabilities are taken to be proportional to the *square root* of the ℓ_2 leverage scores, and have found applications as upper bounds to sensitivities for more general loss functions with less structure than the ℓ_p losses, including the Huber loss and the logistic loss [CW15a, MSSW18, GPV21]. When specialized to ℓ_p subspace embeddings, we obtain the following result by an appropriate modification of our earlier arguments:

Theorem 2.29 (Root leverage score sampling [WY23c])**.** *Let $1 \le p < 2$ and let $\mathbf{A} \in \mathbb{R}^{n\times d}$. Let $\alpha > 0$ and let $p_i = \min\{1, \boldsymbol{\tau}_i^p(\mathbf{A})^{p/2}/\alpha\}$ for $i \in [n]$. Let $\mathbf{S} \in \mathbb{R}^{n\times n}$ be the diagonal matrix formed by independently setting*

$$\mathbf{S}_{i,i} = \begin{cases} \dfrac{1}{p_i^{1/p}} & \text{with probability } p_i \\ 0 & \text{with probability } 1-p_i \end{cases}$$

for each $i \in [n]$. Then, with probability at least $1 - 1/\operatorname{poly}(n)$, there is an α such that $\mathbf{S}$ is an ℓ_p subspace embedding satisfying Definition 2.1 with $\kappa = (1+\varepsilon)$, and furthermore, $\mathbf{S}$ has at most r nonzero rows, for

$$r = \varepsilon^{-2} n^{1-p/2} d^{p/2} \operatorname{poly}\log n.$$

Recursively applying this result gives a matrix $\mathbf{S}$ with

$$r = \varepsilon^{-4/p} d \operatorname{poly}\log n.$$

Note that Theorem 2.29 achieves a nearly optimal dependence on d, while it is slightly loose in the ε dependence.

2.2.3 Active ℓ_p Linear Regression [MMWY22, WY23a]

One of the motivating problems for the study of subspace embeddings is the least squares linear regression problem [DMM06a, Sar06], or more generally, the ℓ_p linear regression problem, in which we wish to solve

$$\min_{\mathbf{x}\in\mathbb{R}^d} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_p^p.$$

When one takes a sampling-based approach to constructing the subspace embedding for the matrix $[\mathbf{A}\ \mathbf{b}]$, including many of the algorithms previously, then the final solution only depends on very few coordinates of the target vector $\mathbf{b}$, namely the r rows sampled by the subspace embedding matrix $\mathbf{S}$. Thus, this gives hope for an algorithm which minimizes the number of entries of the target vector $\mathbf{b}$ it has to read, which is a problem known as *active learning* or *active regression*.

Definition 2.30 (Active ℓ_p linear regression)**.** *An active ℓ_p linear regression algorithm has query complexity r if, given $\mathbf{A} \in \mathbb{R}^{n\times d}$ and query access to the entries of $\mathbf{b} \in \mathbb{R}^n$, it reads r entries of the vectors and outputs $\hat{\mathbf{x}} \in \mathbb{R}^d$ such that*

$$\|\mathbf{A}\hat{\mathbf{x}} - \mathbf{b}\|_p^p \leq (1+\varepsilon) \min_{\mathbf{x}\in\mathbb{R}^d} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_p^p.$$

Our goal is to minimize the query complexity r.

Such an algorithm has significant value in practice, since label acquisition can oftentimes require significantly more resources than the training features, for example if one needs to manually label the training examples to be used.

Unfortunately, the previous approach of constructing sampling-based subspace embeddings for $[\mathbf{A}\ \mathbf{b}]$ does not immediately yield active regression algorithms, since the sampling probabilities will depend on $\mathbf{b}$, and thus the algorithm needs to read all entries of $\mathbf{b}$ anyway. A natural fix is to take the sampling probabilities to only depend on $\mathbf{A}$ but not $\mathbf{b}$, by, for example, using the ℓ_p Lewis weights of the matrix $\mathbf{A}$ without including $\mathbf{b}$. However, the correctness of this algorithm is then no longer clear, as we no longer have the subspace embedding guarantee which includes $\mathbf{b}$. Nonetheless, prior work has shown that this approach in fact *does* yield efficient active regression algorithms in several cases.

For the most important case of $p = 2$, the work of [CP19] obtained an optimal bound of $\Theta(\varepsilon^{-1}d)$, which notably removes a $\log d$ factor that is inherent in sampling-bashed approaches, by using spectral sparsifiers developed in [LS15]. For perhaps the next most important case of $p = 1$, which corresponds to *least absolute deviations regression*, two nearly simultaneous works [CD21, PPP21] showed that a sampling-based approach which takes the sampling probabilities to be the ℓ_1 Lewis weights of $\mathbf{A}$ (without appending $\mathbf{b}$) yields an upper bound of $O(\varepsilon^{-2}d\log(d/\varepsilon))$, with a nearly matching lower bound of $\Omega(\varepsilon^{-2}d)$. However, besides these two special cases, the true sample complexity of active ℓ_p linear regression is far from settled. The only other known bound is an upper bound of $\Omega(\varepsilon^{-2}d^2\log(d/\varepsilon))$ due to [CD21] for $1 < p < 2$. This leads to the following question:

Question 2.31. *What is the query complexity of active ℓ_p linear regression for $p \neq 1, 2$?*

In two works [MMWY22, WY23a], we obtain nearly optimal solutions to Question 2.31 for the entire range of $0 < p < \infty$.

Theorem 2.32 (Nearly optimal active ℓ_p linear regression, [MMWY22, WY23a])**.** *There is an active ℓ_p linear regression algorithm (see Definition 2.30) with query complexity at most r with probability at least $99/100$, where*

$$r = \begin{cases} \tilde{O}(\varepsilon^{-2}d) & 0 < p < 1 \\ \tilde{O}(\varepsilon^{-1}d) & 1 < p < 2 \\ \tilde{O}(\varepsilon^{1-p}d^{p/2}) & 2 < p < \infty \end{cases}$$

Furthermore, for any active ℓ_p linear regression algorithm which succeeds with probability at least $99/100$, its query complexity r must be at least

$$r = \begin{cases} \Omega(\varepsilon^{-2}d) & 0 < p < 1 \\ \Omega(\varepsilon^{-1}d) & 1 < p < 2 \\ \Omega(\varepsilon^{1-p}d^{p/2}) & 2 < p < \infty \end{cases}$$

Notably, we show that there is a sharp phase transition in the behavior of the query complexity at $p = 1$, where $p > 1$ admits an upper bound of $\tilde{O}(\varepsilon^{-1}d)$ queries while $p \leq 1$ requires $\Omega(\varepsilon^{-2}d)$ queries.

The algorithm itself is similar to prior ideas, and we simply take the approach of sampling rows of $\mathbf{A}$ and entries of $\mathbf{b}$ proportionally to the ℓ_p Lewis weights of $\mathbf{A}$. However, the analysis requires significantly new ideas, and in particular, we introduce two key ingredients.

The first is the observation that, while the ℓ_p Lewis weights do not upper bound the sensitivity of the entries of $\mathbf{b}$, any entry $\mathbf{b}_i$ of $\mathbf{b}$ can be classified as either "too big" or "not too big" by comparing $\mathbf{b}_i$ to the ith sensitivity (see Definition 2.25) $\boldsymbol{\sigma}_i(\mathbf{A})$. For entries which are "too big", we show that the loss contribution $|[\mathbf{A}\mathbf{x} - \mathbf{b}](i)|^p = |\langle \mathbf{a}_i, \mathbf{x}\rangle - \mathbf{b}|^p$ on the ith coordinate is dominated by $\mathbf{b}_i$ for any nearly optimal solution $\mathbf{x}$,

and thus this entry can be effectively ignored. On the other hand, for entries $\mathbf{b}_i$ which are "not too big", the sensitivity of $\mathbf{b}_i$ is bounded by $\boldsymbol{\sigma}_i(\mathbf{A})$, which allows an appropriate modification of the chaining arguments for Lewis weight sampling [BLM89, LT91, SZ01] to go through. The idea above is sufficient for nearly optimal bounds for $p < 1$, but for $p > 1$, this still leads to a result that is off by a single ε factor. In order to further optimize our bounds, we additionally introduce a second novel technique which allows us to reduce the ε dependence by using the strict convexity of the ℓ_p loss for $p > 1$. This is done by noting that for $p > 1$, nearly optimal solutions must necessarily be close to the optimal solution, and this fact can be used to improve the sampling error analysis.

Our work has been used to obtain online active regression algorithms in follow-up work of [CLS22].

2.2.4 High-Distortion ℓ_p Subspace Embeddings [WY22a]

Until now, we have focused on subspace embeddings which achieve an distortion of $(1+\varepsilon)$. However, in certain applications, such a high accuracy may not be necessary, and a natural question is whether the number of rows r of the sketch $\mathbf{S}$ can be improved or not if larger errors are allowed. In fact, $(1+\varepsilon)$ is essentially the end of the story of $0 < p \leq 2$, as the upper bounds obtained by ℓ_p Lewis weight sampling (Theorem 2.18) already achieve a bound of $\tilde{O}(\varepsilon^{-2}d)$, and it is easy to see that at least d rows is needed for any subspace embedding, even just to maintain the rank. On the other hand, for $p > 2$, one could still ask for more, since if we require $\Theta(1)$ distortion, then the number of rows necessary is $r = \Omega(d^{p/2})$ [LWW21], whose exponential dependence on p may be prohibitive for large p. In the work of [WY22a], we ask the following question:

Question 2.33. *For $p > 2$, what trade-offs between the number of rows r and the distortion κ are possible in the regime where $\kappa \gg 1$?*

In [WY22a], we provide a nearly optimal trade-off between r and κ as a solution to Question 2.33.

Theorem 2.34 (High-distortion ℓ_p Lewis weight sampling [WY22a])**.** *Let $\mathbf{A} \in \mathbb{R}^{n\times d}$ and $2 < p < \infty$. Then, for any $2 < q < p$, there is a diagonal map $\mathbf{S} \in \mathbb{R}^{n\times n}$ such that, with probability at least $99/100$,*

$$\mathbf{Pr}\Big\{\text{for all } \mathbf{x} \in \mathbb{R}^d, \quad \|\mathbf{A}\mathbf{x}\|_p \leq \|\mathbf{S}\mathbf{A}\mathbf{x}\|_q \leq O(d^{\frac{1}{2}(1-\frac{q}{p})})\|\mathbf{A}\mathbf{x}\|_p\Big\} \geq \frac{99}{100}$$

and furthermore, $\mathbf{S}$ has at most r nonzero rows, for $r = O(d^{q/2}(\log d)^3)$. Furthermore, any randomized algorithm which constructs a data structure $\mathcal{Q}$ such that

$$\mathbf{Pr}\Big\{\text{for all } \mathbf{x} \in \mathbb{R}^d, \quad \|\mathbf{A}\mathbf{x}\|_p \leq \mathcal{Q}(\mathbf{x}) \leq o(d^{\frac{1}{2}(1-\frac{q}{p})})\|\mathbf{A}\mathbf{x}\|_p\Big\} \geq \frac{99}{100}$$

requires $\Omega(d^{q/2+1})$ bits of space.

Our proof of Theorem 2.34 proceeds in two steps: (1) we first show that we can approximate $\|\mathbf{A}\mathbf{x}\|_p$ by $\|\mathbf{W}^{\frac{1}{q}-\frac{1}{p}}\mathbf{A}\mathbf{x}\|_q$ for some diagonal reweighting map $\mathbf{W}$ up to a factor of $d^{\frac{1}{2}(1-\frac{q}{p})}$, and (2) we use ℓ_q Lewis weight sampling to reduce the number of rows to $\tilde{O}(d^{q/2})$ while preserving the distortion up to $\Theta(1)$ factors. Step (2) is simply using Theorem 2.20, so the key ingredient here is step (1).

Perhaps surprisingly, we show that step (1) can in fact also be implemented using ℓ_p Lewis weights, and the reweighting map $\mathbf{W}$ can be simply be taken to be the ℓ_p Lewis weights. More specifically, we show the following theorem:

Theorem 2.35 (ℓ_p Lewis weight change of density [WY22a])**.** *Let $\mathbf{A} \in \mathbb{R}^{n\times d}$ and let $0 < q < p < \infty$. Let $\mathbf{W} = \operatorname{diag}(\mathbf{w}^p(\mathbf{A}))$ be the diagonal map given by the ℓ_p Lewis weights of $\mathbf{A}$. Then, there is a scaling factor c such that for every $\mathbf{x} \in \mathbb{R}^d$,*

$$\|\mathbf{A}\mathbf{x}\|_p \leq c\|\mathbf{W}^{\frac{1}{q}-\frac{1}{p}}\mathbf{A}\mathbf{x}\|_q \leq \kappa\|\mathbf{A}\mathbf{x}\|_p$$

for

$$\kappa = \begin{cases} d^{\frac{1}{q}-\frac{1}{p}} & \min(p,q) \leq 2 \\ d^{\frac{1}{2}(1-\frac{q}{p})} & \min(p,q) \geq 2 \end{cases}$$

In fact, the result of Theorem 2.35 provides an elementary proof of a result of [LT80] from the geometric functional analysis literature, who proved the existence of a diagonal map satisfying the guarantees of Theorem 2.35 by using sophisticated results from the theory of factorization of operators, p-summing norms, and operator ideals. On the other hand, our proof of Theorem 2.35 only requires elementary inequalities and ℓ_p Lewis weights. One of the key insights we use is that if $\mathbf{W}$ are the ℓ_p Lewis weights, then the $\mathbf{W}$ is also the ℓ_q Lewis weights of the matrix $\mathbf{W}^{\frac{1}{q}-\frac{1}{p}}\mathbf{A}$.

2.2.5 Streaming ℓ_∞ Subspace Embeddings and Computational Geometry [WY22a]

An investigation of high-distortion ℓ_p subspace embeddings for $p > 2$ prompts a closely related study in the *streaming setting*, in which we must compute an ℓ_p subspace embedding of the matrix $\mathbf{A} \in \mathbb{R}^{n\times d}$, when $\mathbf{A}$ is presented as n rows $\mathbf{a}_i \in \mathbb{R}^d$ which arrive one by one in one pass over a stream.

Definition 2.36 (Geometric streaming setting)**.** *In the geometric streaming setting, an algorithm receives as input an integer matrix $\mathbf{A} \in \mathbb{Z}^{n\times d}$ with entries bounded by $|\mathbf{A}_{i,j}| \in [-\operatorname{poly}(n), \operatorname{poly}(n)]$ in a row arrival stream, that is, the algorithm sees the rows $\mathbf{a}_i \in \mathbb{Z}^d$ of $\mathbf{A}$ one at a time in one pass through the stream.*

Note that when we have algorithms for subspace embeddings with $(1+\varepsilon)$ distortion, then we can easily obtain a streaming algorithm by a technique known as *merge-and-reduce*, in which we iteratively perform the operations of concatenating new rows and reducing the size of the stored subspace embedding by re-computing a subspace embedding. These operations can be performed in a way such that the subspace embedding is re-computed at a "depth" of only $O(\log n)$ if the input matrix $\mathbf{A}$ has n rows, meaning that if we compute subspace embeddings with distortion $(1+\varepsilon/\log n)$ at each step, then the total distortion is only $(1+\varepsilon/\log n)^{\log n} = (1+O(\varepsilon))$. However, this trick does not work for when our distortions are $\kappa = (1+\Omega(1))$, and leads to $\operatorname{poly}(n)$ factor total distortions when applied in this case.

Perhaps the most important case of this problem is that of computing ℓ_∞ subspace embeddings in the streaming model. In this case, Theorem 2.35, both in the upper bound and lower bound, can be generalized to show that ℓ_∞ subspace embeddings with $\kappa = \sqrt{d}$ distortion and $r = d$ rows can be obtained, and that the upper bound comes from ℓ_∞ Lewis weights, which corresponds to the well-studied problem of *Löwner–John ellipsoids* [Joh48, Tod16], also known as minimum volume enclosing ellipsoids. However, the question of computing Löwner–John ellipsoids in the streaming setting using only $\operatorname{poly}(d)$ bits of space is a central unresolved problem in the literature of computational geometry [MSS10, AS15]. Indeed, the only known prior results for computing Löwner–John ellipsoids in a stream uses $\exp(\operatorname{poly}(d))$ bits of space in order to estimate the extent of every direction in $\mathbb{R}^d$ using a net [AHV04, AHV05], rather than polynomial in d. Thus the question of efficiently maintaining ℓ_∞ subspace embeddings in a stream is an important problem.

In fact, in our work of [WY22a], we resolve both the problem of maintaining ℓ_∞ subspace embeddings and Löwner–John ellipsoids in the streaming setting, and in fact, a multitude of other problems in the streaming computational geometry literature which previously only admitted upper bounds with exponential dependencies in the dimension. Our central theorem is the following:

Theorem 2.37 (Streaming ℓ_∞ subspace embedding [WY22a])**.** *There is a deterministic streaming algorithm such that, for any $\mathbf{A} \in \mathbb{Z}^{n\times d}$ presented in a geometric stream (Definition 2.36), the algorithm maintains $\mathbf{SA}$ for a matrix $\mathbf{S} \in \mathbb{Z}^{r\times n}$ such that for every $\mathbf{x} \in \mathbb{R}^d$,*

$$\|\mathbf{Ax}\|_\infty \le \|\mathbf{SAx}\|_\infty \le O(\sqrt{d\log n})\|\mathbf{Ax}\|_\infty.$$

Furthermore, the algorithm uses at most $O(d^2(\log n)^2)$ bits of space.

Our main technique is the use of *online leverage scores*, which were introduced by [CMP16, CMP20], as a tool both to discover directions $\mathbf{x} \in \mathbb{R}^d$ in which the ℓ_∞ norm $\|\mathbf{Ax}\|_\infty$ is updated significantly in a stream, and to bound the total number of such updates which can occur. Our work also shows how to sharpen the bound on the sum of online leverage scores given by [CMP16, CMP20] when the input matrix is an integer matrix with bounded bit complexity, which answers open questions asked in [BDM+20].

A related result on maintaining Löwner–John ellipsoids in the streaming setting has been obtained in concurrent work of [MMO22], which achieve results that depend on a certain condition number of the ellipsoid.

2.2.6 Subspace Embeddings for General Losses [MMWY22]

Up until now, we have conducted a thorough study of subspace embeddings for the ℓ_p loss, with applications to ℓ_p regression in mind. In fact, the problem of computing subspace embeddings makes sense in a far more generalized setting, where we wish to approximate loss functions of the form

$$\|\mathbf{A}\mathbf{x}\|_{g,\mathbf{w}} \coloneqq \sum_{i=1}^{n} \mathbf{w}_i \cdot g([\mathbf{A}\mathbf{x}](i)), \tag{3}$$

where we denote the loss function as a norm in an abuse of notation, despite the fact that $\|\cdot\|_{g,\mathbf{w}}$ may not be a norm. For example, taking the weights $\mathbf{w}_i$ to be all ones and g to be the so-called *Huber loss* H defined as

$$H(x) \coloneqq \begin{cases} x^2/2 & |x| \le 1 \\ |x| - 1/2 & |x| \ge 1 \end{cases}$$

is useful in solving linear regression with the Huber loss, which is a popular loss function in the literature of robust statistics [CW15a]. Similarly, taking g to be the *Tukey loss* T defined as

$$T(x) \coloneqq \begin{cases} 1 - (1 - x^2)^3 & |x| \le 1 \\ 1 & |x| \ge 1 \end{cases}$$

is another popular choice for robust regression [CWW19]. Yet another example is to take g to be the *logistic loss*, given by

$$g(x) \coloneqq \log(1 + e^x)$$

which corresponds to logistic regression [MSSW18, MMR21].

Improved sensitivity bounds for general loss functions. In fact, we have already discussed a generalized approach to estimating functions of the form of (3) in Section 2.2.2, via *sensitivity sampling*. Recall that in this framework, we wish to compute upper bounds on the sensitivity scores $\boldsymbol{\sigma}_i$, which in this case are given by

$$\boldsymbol{\sigma}_i(\mathbf{A}) \coloneqq \sup_{\mathbf{A}\mathbf{x} \neq 0} \frac{\mathbf{w}_i \cdot g([\mathbf{A}\mathbf{x}](i))}{\sum_{j=1}^{n} \mathbf{w}_j \cdot g([\mathbf{A}\mathbf{x}](j))}.$$

Given upper bounds $\tilde{\boldsymbol{\sigma}}_i \ge \boldsymbol{\sigma}_i(\mathbf{A})$ on the sensitivity scores, we fairly immediately obtain a sampling algorithm which samples at most $\widetilde{O}(\varepsilon^{-2}\tilde{\mathfrak{S}}d)$ rows of $\mathbf{A}$, where $\tilde{\mathfrak{S}} = \sum_{i=1}^{n} \tilde{\boldsymbol{\sigma}}_i$. The primary difficult in this approach is efficiently obtaining the sensitivity upper bounds $\tilde{\boldsymbol{\sigma}}_i$. Previously, an approach based on ellipsoidal rounding of the balls induced by the norm $\|\mathbf{A}\mathbf{x}\|_{g,\mathbf{w}}$ has been proposed by [TMF20]. However, computing Löwner–John ellipsoids for general convex bodies is computational expensive, and furthermore, leads to $\mathrm{poly}(d)$ factor losses in the total sensitivity upper bound $\tilde{\mathfrak{S}}$ and thus in the sample complexity.

In the work of [MMWY22], we obtain a significantly improved algorithm for estimating sensitivity scores, which is nearly optimal for a wide class of loss functions.

Theorem 2.38 (Sensitivity upper bounds for general loss functions, Theorem 4.9, [MMWY22])**.** *Let $M : \mathbb{R}_{\ge 0} \to \mathbb{R}_{\ge 0}$ be increasing, has $M(0) = 0$, and has at most quadratic growth, that is,*

$$\frac{M(y)}{M(x)} \le c\Big(\frac{y}{x}\Big)^2$$

for all $y > x$. Let $g(x) \coloneqq M(|x|)$. Then, there is an algorithm that computes upper bounds $\tilde{\boldsymbol{\sigma}}_i$ to the sensitivities with respect to g such that $\tilde{\mathfrak{S}} = \sum_{i=1}^{n} \boldsymbol{\sigma}_i \le O(d \log^2 n + \tau)$ in time

$$O\bigg(\mathsf{nnz}(\mathbf{A}) \log^3 n + \frac{nd^\omega}{\tau} \log n\bigg).$$

The class of functions handled by Theorem 2.38 include the Huber loss, any ℓ_p loss for $p \leq 2$, as well as a wide variety of loss functions considered in the robust statistics literature that behave similarly to the Huber loss, that is, quadratic growth near the origin and linear growth away from the origin.

The main idea towards obtaining Theorem 2.38 is starts from an observation from the streaming literature [BO10] that for functions g of at most quadratic growth, entries $i \in [n]$ of a vector $\mathbf{y}$ which are "heavy" in the g loss, that is, $g(\mathbf{y}_i)/\|\mathbf{y}\|_g = \Omega(1)$, must necessarily be "heavy" in the ℓ_2 loss. Thus, a superset of heavy elements in the g loss can be identified by identifying the heavy elements in the ℓ_2 loss, and furthermore, this superset is not too large by the definition of heaviness. This can then be generalized to identifying ε-heavy elements, that is, $g(\mathbf{y}_i)/\|\mathbf{y}\|_g \geq \varepsilon$, based on a trick by randomly hashing the entries of $\mathbf{y}$ into $O(1/\varepsilon)$ buckets so that, within this bucket, an ε-heavy entry is likely to be $\Omega(1)$-heavy. We can now draw an analogy between "heavy" entries under the g loss with rows of $\mathbf{A}$ with large sensitivity $\boldsymbol{\sigma}_i$, as well as "heavy" entries under the ℓ_2 loss with rows of $\mathbf{A}$ with large ℓ_2 leverage score. Thus, by combining leverage score estimation with a hashing trick, we arrive at our Theorem 2.38.

Sharper sample complexity for the Huber loss. As we previously observed in Section 2.2.2, the sensitivity sampling framework, when applied naïvely, is loose for certain problems such as ℓ_2 subspace embeddings. Thus, one may hope for even sharper results than those obtained as a consequence of sensitivity sampling combined with Theorem 2.38. In [MMWY22], we also study the problem of whether the sample complexity bound of $\tilde{O}(\varepsilon^{-2}d^2\log^2 n)$ from the sensitivity sampling approach can be improved for the Huber loss, which has more structure than general loss functions handled in Theorem 2.38. Indeed, we are able to obtain such a result:

Theorem 2.39 (Subspace embeddings for the Huber loss [MMWY22])**.** *Let $\mathbf{A} \in \mathbb{R}^{n\times d}$ and $d = \Omega(\log(1/\varepsilon))$. Then, there is an algorithm which outputs weights $\mathbf{w} \in \mathbb{R}^n_{\geq 0}$ such that*

$$\mathbf{Pr}\left\{\sum_{i=1}^n H([\mathbf{Ax}](i)) = (1\pm\varepsilon)\sum_{i=1}^n \mathbf{w}_i \cdot H([\mathbf{Ax}](i)\right\} \geq \frac{99}{100}$$

and furthermore, $\mathbf{w}$ has at most r nonzero entries, for $r = d^{4-2\sqrt{2}}\cdot \mathrm{poly}(\log n, \varepsilon^{-1})$, where $4-2\sqrt{2} < 1.172$.

Recall that for ℓ_p subspace embeddings, one of the crucial ingredients towards obtaining nearly optimal bounds of $\tilde{O}(\varepsilon^{-2}d)$ for $p \leq 2$ and $\tilde{O}(\varepsilon^{-2}d^{p/2})$ for $p > 2$ is a chaining argument developed in the works of [BLM89, LT91, SZ01]. However, the net constructions in these arguments are tailored to ℓ_p losses, and heavily make use of special algebraic properties of ℓ_p losses which are not available to the Huber loss. Instead, we show that in order to develop chaining arguments for the Huber loss, we can in fact directly use the net constructions for ℓ_p losses, if we oversample by an appropriate factor. Furthermore, we further optimize this argument by showing that at each "radius" of the Huber loss $\|\mathbf{Ax}\|_H$, we can use a different choice of $p \in [1, 2]$ depending on which choice of p yields the smallest distortion.

2.2.7 Future Directions for Non-Oblivious Subspace Embeddings

Many important directions remain open in the study of subspace embeddings, both for ℓ_p losses as well as general losses.

Nearly Optimal Bounds for ℓ_p Subspace Embeddings for $p > 2$. One of the outstanding gaps in bounds for ℓ_p subspace embeddings is the optimality of the upper bound given in Theorem 2.15 and Theorem 2.20 in terms of the dependence on d and ε for $p > 2$. So far, the upper bound is $r = \tilde{O}(\varepsilon^{-2}d^{p/2})$ for a subspace embedding $\mathbf{S}$ with r rows, while the best known lower bound is still Theorem 2.16 due to [LWW21], which gives a lower bound of $r = \tilde{\Omega}(\varepsilon^{-1}d^{p/2} + \varepsilon^{-2}d)$. Thus, resolving this last gap from obtaining nearly optimal trade-offs between number of rows r, d, and the accuracy parameter ε is our first open question about ℓ_p subspace embeddings.

Question 2.40. *For $p \in (2,\infty)\setminus 2\mathbb{Z}$, what is the smallest possible number of rows r that is possible for ℓ_p subspace embeddings with $(1+\varepsilon)$ distortion? Is there a lower bound showing that $r = \Omega(\varepsilon^{-2}d^{p/2})$ rows is necessary?*

Deterministic ℓ_p Subspace Embeddings. For $p = 2$, the seminal work of [BSS09, BSS12] showed that it is possible to deterministically obtain ℓ_2 subspace embeddings with $r = O(\varepsilon^{-2}d)$ rows in polynomial time, and has spurred multiple works further improving the running time of this algorithm [Zou12, ALO15]. This algorithm, however, makes heavy use of the special structure of the ℓ_2 norm, and does not yield results for ℓ_p subspace embeddings for $p \neq 2$. Thus, an interesting question is whether polynomial time algorithms for constructing ℓ_p subspace embeddings exist or not.

Question 2.41. *Is there a deterministic polynomial time algorithm for constructing $(1+\varepsilon)$-approximate ℓ_p subspace embeddings with $\tilde{O}(\varepsilon^{-2}d)$ rows for $p < 2$ or $\tilde{O}(\varepsilon^{-2}d^{p/2})$ rows for $p > 2$?*

In fact, even a Las Vegas algorithm for computing ℓ_p subspace embeddings may be interesting, as there are currently no known efficient algorithms for checking whether two matrices are close in the sense of ℓ_p subspace embeddings, for any $p \neq 2$:

Question 2.42. *Is there a polynomial time Las Vegas algorithm for constructing $(1+\varepsilon)$-approximate ℓ_p subspace embeddings with $\tilde{O}(\varepsilon^{-2}d)$ rows for $p < 2$ or $\tilde{O}(\varepsilon^{-2}d^{p/2})$ rows for $p > 2$?*

Removing Logarithmic Factors for ℓ_p Subspace Embeddings. A closely related problem to Question 2.41 is the question of removing logarithmic factors in the number of rows r. In particular, the work of [BSS09, BSS12] as well as its various follow-ups [Zou12, ALO15, LS15] obtain $r = O(\varepsilon^{-2}d)$, without any logarithmic factor losses. On the other hand, for independent sampling-based approaches such as Lewis weight sampling, an extra logarithmic factor is inherent due to the coupon-collector problem. However, for most values of $p \neq 2$[1], no other approaches towards obtaining $(1+\varepsilon)$-approximate ℓ_p subspace embeddings are known. Thus, an important question is the following:

Question 2.43. *Is there an algorithm for constructing $(1+\varepsilon)$-approximate ℓ_p subspace embeddings with $r = O(\varepsilon^{-2}d)$ rows for $p < 2$ and $r = O(\varepsilon^{-2}d^{p/2})$ rows for $p > 2$?*

For $p = 1$, this problem has been raised in [Sch07, HRR22].

Subspace Embeddings for the Huber Loss. In our Theorem 2.39 from [MMWY22], we have made substantial progress in obtaining sharper bounds for subspace embeddings for the Huber loss, showing that the dependence on d can be reduced to $d^{4-2\sqrt{2}}$, where $4 - 2\sqrt{2} < 1.172$. An important question is whether this d dependence can be reduced all the way down to d or not. This question has applications beyond Huber regression, and can be used in subroutines for fast algorithms for high precision algorithms for ℓ_p regression [APS19, AKPS19, AS20, GPV21].

Question 2.44. *Is there an algorithm for constructing $(1+\varepsilon)$-approximate subspace embeddings for the Huber loss with $r = d \cdot \mathrm{poly}(\log n, \varepsilon^{-1})$ rows?*

One promising approach to this problem is the root leverage score sampling algorithm, which has used in [CW15a, GPV21] to obtain Huber subspace embeddings, and was shown to yield bounds of the form $d \cdot \mathrm{poly}(\log n, \varepsilon^{-1})$ in [WY23c] for the ℓ_p loss.

Nearly Optimal Guarantees for Sensitivity Sampling. Finally, we re-iterate our main open question, Question 2.27, from the work of [WY23c] from Section 2.2.2: what is the smallest sample complexity possible for the ℓ_p sensitivity sampling algorithm? While we have achieved the bounds of $\tilde{O}(\varepsilon^{-2}\mathfrak{S}^{2/p})$ for $p < 2$ and $\tilde{O}(\varepsilon^{-2}\mathfrak{S}^{2-2/p})$ for $p > 2$, we conjecture that a bound of $\tilde{O}(\varepsilon^{-2}(\mathfrak{S} + d))$ is possible.

3 Low Rank Approximation

Along with subspace embeddings and linear regression, *low rank approximation*, which is the problem of approximating matrices by one of lower rank, is one of the foundational problems in the field of randomized numerical linear algebra [FKV04, DV06, DKM06a, DKM06b, DKM06c, DMM06b].

[1] An important exception is $p \in 2\mathbb{Z}$, which admit exact isometries via other methods due to its special structure [Sch11].

Definition 3.1. *In the general rank k approximation problem, we consider the problem of finding a rank k matrix $\hat{\mathbf{A}}$ (or a rank r matrix with r slightly larger than k) such that*

$$\|\mathbf{A} - \hat{\mathbf{A}}\| \leq \kappa \min_{\text{rank}(\mathbf{A}') \leq k} \|\mathbf{A} - \mathbf{A}'\|.$$

3.1 Column Subset Selection with Entrywise Losses [WY23a]

As is the case for subspace embeddings and linear regression, the problem of low rank approximation is best understood when the norm under consideration is the ℓ_2 loss (which corresponds to the Frobenius norm in this case), and a long line of work has studied fast randomized algorithms for low rank approximation under the Frobenius norm [FKV04, DV06, DKM06a, DKM06b, DKM06c, DMM06b, CW13, MM15, CMM17, BW17]. However, when the input matrix is corrupted by heavy-tailed noise or include outliers, the ℓ_2 norm is not always the most desirable due to the fact that it tends to fit to the outliers too much. Thus, oftentimes, it is desirable to solve the low rank approximation problem under other error measures, especially those with slower growth than the ℓ_2 loss. One notable class of losses is the *entrywise ℓ_p loss*, and more generally, the *entrywise g loss*, where g can be an arbitrary loss function.

Definition 3.2 (Entrywise losses)**.** *Let $\mathbf{A} \in \mathbb{R}^{n \times d}$ and let $g : \mathbb{R} \to \mathbb{R}_{\geq 0}$. Then, we define the* entrywise g norm *of $\mathbf{A}$ as*

$$\|\mathbf{A}\|_g \coloneqq \sum_{i=1}^{n} \sum_{j=1}^{d} g(\mathbf{A}_{i,j}).$$

When $g(x) = |x|^p$, then we instead define

$$\|\mathbf{A}\|_{p,p} \coloneqq \left(\sum_{i=1}^{n} \sum_{j=1}^{d} |\mathbf{A}_{i,j}|^p \right)^{1/p}$$

to be the entrywise ℓ_p loss.

For $p \neq 2$, the entrywise loss low rank approximation is hard to approximate in a variety of settings [Mie09, GV18, DHJ+18, BBB+19, MW21] and thus we need to allow for an appropropriate notion of approximation. We study bicriteria approximation guarantees of the following form:

Definition 3.3 (Bicriteria Coreset for Low Rank Approximation)**.** *Let $\mathbf{A} \in \mathbb{R}^{n \times d}$, let k be a rank parameter, and let $\|\cdot\|$ be any loss function. Let $S \subseteq [d]$ a subset of columns, and write $\mathbf{A}|^S$ for the $n \times S$[2] matrix formed by the columns of $\mathbf{A}$ indexed by S. Then, S is a* bicriteria coreset *with distortion $\kappa \geq 1$ if*

$$\min_{\mathbf{X} \in \mathbb{R}^{S \times d}} \|\mathbf{A} - \mathbf{A}|^S \mathbf{X}\| \leq \kappa \min_{\text{rank}(\mathbf{A}') \leq k} \|\mathbf{A} - \mathbf{A}'\|.$$

3.1.1 Algorithms for General Entrywise Losses

We begin by presenting our result on entrywise g-norm low rank approximation, first considered by [SWZ19]. For our analysis, we will need to assume several natural properties on g, which have been considered in previous work [CW15b, CW15a, SWZ19, MMWY22] for obtaining provable guarantees for randomized numerical linear algebra under a broad class of loss functions:

Definition 3.4. *Let $g : \mathbb{R} \to \mathbb{R}_{\geq 0}$. Then:*

- *g satisfies the* $\mathsf{ati}_{g,t}$-approximate triangle inequality *if for any $x_1, x_2, \ldots, x_t$, $g(\sum x_i) \leq \mathsf{ati}_{g,t} \cdot \sum_i g(x_i)$.*
- *g is* mon_g-monotone *if for any $0 \leq |x| \leq |y|$, $g(x) \leq \mathsf{mon}_g \cdot g(y)$.*
- *g has at least* lin_g-linear growth *if for any $0 < |x| \leq |y|$, $g(y)/g(x) \geq \mathsf{lin}_g \cdot |y|/|x|$.*

[2] We allow for indexing matrices and vectors by arbitrary sets. For example, $\mathbb{R}^S$ is the set of vectors with entries indexed by elements s of S, and $\mathbb{R}^{S \times d}$ is the set of matrices with rows indexed by elements of S and columns indexed by $[d]$.

For example, popular functions that satisfy these bounds include the Huber loss, Fair loss, Cauchy loss, ℓ_1-ℓ_2 loss, and the quantile loss [SWZ19]. While the lin_g-linear growth bound excludes the Tukey loss, which grows quadratically near the origin and stays constant away from the origin, it allows for a modification of the Tukey loss where the constant away from the origin is replaced by an arbitrarily slow linear growth [CW15a].

[SWZ19] showed that, given an algorithm for solving linear regression in the g-norm with relative error reg_g, it is possible to compute a set of $O(k \log d)$ columns achieving an approximation ratio of

$$O(k \log k) \cdot \mathsf{reg}_g \cdot \mathsf{mon}_g \cdot \mathsf{ati}_{g,k+1}.$$

for g satisfying the mon_g-monotone and $\mathsf{ati}_{g,t}$-approximate triangle inequality properties. We show that for the slightly restricted family of g of at least lin_g-linear growth, which for example includes all convex g [CW15a], we obtain an improved approximation ratio of

$$O(\sqrt{k \log \log k}) \cdot \frac{\mathsf{reg}_g \cdot \mathsf{ati}_{g,s+1}}{\mathsf{lin}_g}.$$

Our guarantee matches, and in fact improves a log factor, of the ℓ_1 column subset selection guarantee of [MW21], despite being a far more general result. Furthermore, our bound is tight, in the sense that the $\sqrt{k}$ cannot be improved to a smaller polynomial due to a matching lower bound for ℓ_1 column subset selection [SWZ17]. Our technique for removing the $\log k$ factor in the distortion is general, and can be used to improve prior results for ℓ_p column subset selection as well [CGK+17, DWZ+19, MW21].

Theorem 3.5 (Improved guarantees for entrywise low rank approximation)**.** *Let $\mathbf{A} \in \mathbb{R}^{n \times d}$ and let $k \geq 1$. Let $s = O(k \log \log k)$. Let $g : \mathbb{R} \to \mathbb{R}_{\geq 0}$ be a loss function satisfying the $\mathsf{ati}_{g,t}$-approximate triangle inequality for $t = s + 1$ and the lin_g-linear growth property. Furthermore, suppose that there is an algorithm outputting $\tilde{\mathbf{x}}$ such that*

$$\|\mathbf{B}\tilde{\mathbf{x}} - \mathbf{b}\|_g \leq \mathsf{reg}_{g,s} \cdot \min_{\mathbf{x} \in \mathbb{R}^s} \|\mathbf{B}\tilde{\mathbf{x}} - \mathbf{b}\|_g$$

for any $\mathbf{B} \in \mathbb{R}^{n \times s}$ and $\mathbf{b} \in \mathbb{R}^n$. Then, there is an algorithm which outputs a subset $S \subseteq [d]$ of $|S| = O(k(\log \log k)(\log d)^2)$ columns and $\mathbf{X} \in \mathbb{R}^{t \times d}$ such that

$$\left\|\mathbf{A} - \mathbf{A}|^S \mathbf{X}\right\|_g \leq O(\sqrt{s}) \frac{\mathsf{reg}_{g,O(s \log d)} \cdot \mathsf{ati}_{g,s+1}}{\mathsf{lin}_g} \min_{\mathrm{rank}(\mathbf{A}') \leq k} \|\mathbf{A} - \mathbf{A}'\|_g.$$

For the important case of the Huber loss, given by

$$H(x) = \begin{cases} |x|^2/2 & \text{if } |x| \leq 1 \\ |x| - 1/2 & \text{if } |x| > 1 \end{cases},$$

we specialize our technique to give the following optimized result:

Theorem 3.6 (Entrywise Huber Low Rank Approximation)**.** *Let $\mathbf{A} \in \mathbb{R}^{n \times d}$ and let $k \geq 1$. There is an algorithm which outputs a subset $S \subseteq [d]$ of $|S| = O(k(\log \log k) \log d)$ columns and $\mathbf{X} \in \mathbb{R}^{S \times d}$ such that*

$$\left\|\mathbf{A} - \mathbf{A}|^S \mathbf{X}\right\|_H \leq O(k) \min_{\mathrm{rank}(\mathbf{A}') \leq k} \|\mathbf{A} - \mathbf{A}'\|_H,$$

where $\|\cdot\|_H$ denotes the entrywise Huber loss.

The previous best known bound [SWZ19] gave a distortion of $\tilde{O}(k^2)$ for the same number of columns.

For both general entrywise low rank approximation as well as low rank approximation under the Huber loss, our new results are in fact based on our new constructions of well-conditioned spanning sets in Theorem 2.8.

3.1.2 Algorithms for the Entrywise ℓ_p Norm

For $p \neq 2$, efficient bicriteria approximations for entrywise ℓ_p low rank approximation were obtained in a line of work initiated by [SWZ17], who studied the case of $p = 1$. For other $p \neq 2$, [CGK+17, DWZ+19] gave algorithms selecting $O(k \log d)$ columns achieving a distortion of $\tilde{O}(k^{1/p})$ for $p < 2$ and $\tilde{O}(k^{1-1/p})$ for $p > 2$, and a hardness result showing that any approximation spanned by k columns must have distortion at least

$$\Omega(k^{1-1/p}) \tag{4}$$

Perhaps surprisingly, [MW21] then showed that the lower bound of (4) could be circumvented when $p < 2$, by giving an algorithm which selected $\tilde{O}(k \log d)$ columns and achieved a distortion of $\tilde{O}(k^{1/p-1/2})$. Note that this does not contradict the lower bound, since the hardness result of (4) applies only when *exactly* k columns are selected. It was also shown that this result was optimal for such bicriteria algorithms, with a lower bound ruling out $k^{1/p-1/2-o(1)}$ approximations for any algorithm selecting $\tilde{O}(k)$ columns, based on a result of [SWZ17] which ruled out $k^{1/2-o(1)}$ approximations for any set of $\text{poly}(k)$ columns for $p = 1$.

Unfortunately, the algorithmic result of [MW21] uses p-stable random variables [Nol20] which only exist for $p \leq 2$, and similar improvements were not given for $p > 2$. Similarly, the hardness results also rely on specific properties of $p < 2$, and do not apply to $p > 2$. This motivates the following question:

Question 3.7. *What distortions are possible for entrywise ℓ_p low rank approximation, if $O(k \log d)$ columns can be selected?*

Our main result for entrywise ℓ_p low rank approximation is an algorithm which achieves the natural analogue of the algorithmic result of [MW21], which circumvents (4):

Theorem 3.8 (Entrywise ℓ_p low rank approximation [WY23a])**.** *Let $p \in [2, \infty]$, let $\mathbf{A} \in \mathbb{R}^{n \times d}$, and let $k \geq 1$. There is an algorithm which outputs a subset $S \subseteq [d]$ of $O(k \log d)$ columns and $\mathbf{X} \in \mathbb{R}^{S \times d}$ such that*

$$\left\|\mathbf{A} - \mathbf{A}|^S \mathbf{X}\right\|_{p,p} \leq O(k^{1/2-1/p}) \min_{\text{rank}(\mathbf{A}') \leq k} \|\mathbf{A} - \mathbf{A}'\|_{p,p}.$$

For $p = \infty$, we show that Theorem 3.8 is tight by showing that any set of at most $\text{poly}(k)$ columns cannot achieve a distortion better than $k^{1/2-o(1)}$.

3.2 Online Subspace Approximation [WY23a]

In addition to the generalization of the Frobenius norm to general entrywise losses considered in Section 3.1, another matrix loss for low rank approximation that is often considered is the $(p, 2)$-loss, which takes the ℓ_2 norms of the n rows matrix, and then takes the ℓ_p rows of the resulting n numbers.

Definition 3.9. *Let $\mathbf{A} \in \mathbb{R}^{n \times d}$. Then, we define the $(p, 2)$-norm of $\mathbf{A}$ as*

$$\|\mathbf{A}\|_{p,2} \coloneqq \left(\sum_{i=1}^{n} \left\|\mathbf{e}_i^\top \mathbf{A}\right\|_2^p \right)^{1/p}.$$

Under this loss, the low rank approximation problem has a more geometric intuition due to the structure of the ℓ_2 norm. Indeed, it can be shown that the optimal rank k approximation $\mathbf{A}^*$ to $\mathbf{A}$ under the $(p, 2)$-norm loss takes the form of $\mathbf{A}^* = \mathbf{AP}$ for some orthogonal projection matrix $\mathbf{P}$ onto a k-dimensional subspace. Thus, it suffices to minimize over only matrices of the form $\mathbf{AP}$, and in this case, the loss can be viewed as the ℓ_p norms of the distances when projecting the n rows $\{\mathbf{a}_i\}_{i=1}^n$ of $\mathbf{A}$ onto the subspace spanned by $\mathbf{P}$. This is known as the ℓ_p *subspace approximation* problem.

Definition 3.10 (ℓ_p subspace approximation)**.** *Let $\mathcal{F}_k$ denote the set of subspaces $F \subseteq \mathbb{R}^d$ of rank at most k. We seek a rank k subspace $F \in \mathcal{F}_k$ which approximately minimizes*

$$\|\mathbf{A} - \mathbf{AP}_F\|_{p,2} = \left[\sum_{i=1}^{n} \|\mathbf{a}_i - \mathbf{P}_F \mathbf{a}_i\|_2^p \right]^{1/p} = \left[\sum_{i=1}^{n} \min_{\mathbf{x} \in F} \|\mathbf{a}_i - \mathbf{x}\|_2^p \right]^{1/p},$$

where $\mathbf{a}_i = \mathbf{e}_i^\top \mathbf{A}$ and $\mathbf{P}_F$ is the orthogonal projection matrix onto F.

As with entrywise low rank approximation, the ℓ_p subspace approximation problem is computationally hard for $p \neq 2$ [DTV11, GRSW12, CW15b]. For ℓ_p subspace approximation, a particularly fruitful approach towards designing approximation algorithms is by designing *coresets* for this problem, in which we select a weighted subset S of the input points $\mathbf{a}_i$ to approximate the objective function of Definition 3.10.

Definition 3.11 (Strong coreset)**.** *Let $\mathbf{A} \in \mathbb{R}^{n\times d}$, let $p \geq 1$, and let $k \geq 1$ be a rank parameter. Then, a subset $S \subseteq [n]$ together with weights $\mathbf{w} \in \mathbb{R}^S$ is a* strong coreset *if*

$$\text{for all } F \in \mathcal{F}_k, \qquad \sum_{i=1}^{n} \|\mathbf{a}_i - \mathbf{P}_F \mathbf{a}_i\|_2^p = (1 \pm \varepsilon) \sum_{i \in S} \mathbf{w}_i \|\mathbf{a}_i - \mathbf{P}_F \mathbf{a}_i\|_2^p. \tag{5}$$

In this setting, it is possible to reduce the number of points to just $\text{poly}(k/\varepsilon)$, and sensitivity sampling (see Section 2.2.2) [LS10, FL11, VX12, HV20] as well as other sampling-based approaches [DV07] have been studied for this problem.

For subspace approximation, it turns out that the study of *online coresets* is particularly interesting, where the input points $\{\mathbf{a}_i\}_{i=1}^n$ arrive one by one in one pass through a stream, and for each $i \in [n]$, we must irrevocably choose whether to include $\mathbf{a}_i$ in the coreset or permanently discard it. For $p = 2$, the problem of designing online coresets was studied in the works of [BLVZ19, BDM+20], which achieves a nearly optimal bound of approximately $\tilde{O}(k/\varepsilon^2)$ vectors, up to a necessary logarithmic factor dependence on the "condition number" of the stream, based on the ridge leverage score sampling algorithm of [CMM17]. However, for general p, the analogous problem is significantly more difficult, due to the fact that existing sampling-based algorithms for ℓ_p subspace approximation all require a "two-stage" approach, in which a crude approximation is first computed, and then used to re-sample a coreset.

In the work of [WY23a], we show that sensitivity sampling can in fact be implemented so that the two stages of sensitivity sampling can be implemented in an online fashion, leading to the first online coresets for ℓ_p subspace approximation.

Theorem 3.12 (Online coreset for ℓ_p subspace approximation)**.** *Let $\mathbf{A} \in \mathbb{R}^{n\times d}$ have online condition number $\kappa^{\mathsf{OL}} := \|\mathbf{A}\|_2 \max_{i=1}^n \|\mathbf{A}_i^-\|_2$[3], $\varepsilon \in (0,1)$, $p \geq 1$ a constant, and let k be a rank. There is an online coreset algorithm, which, with probability at least $99/100$, stores a weighted subset of rows S with weights $\mathbf{w} \in \mathbb{R}^S$ satisfying (5) such that, for $\varepsilon' = \varepsilon^{(p+3)\cdot(1\vee(2/p))}$*[4]*,*

$$|S| = \begin{cases} O\big(k^2\big(\varepsilon'^{-2} + \varepsilon^{-2}\varepsilon'^{-1}k^2\big)\big)\log(n\kappa^{\mathsf{OL}})^{O(1)} & \text{if } p < 2 \\ O\Big(k^p\Big(k^{p/2+1} + \varepsilon'^{-2} + \varepsilon^{-2}\varepsilon'^{-1}k^2\Big)\Big)\log(n\kappa^{\mathsf{OL}})^{O(1)} & \text{if } 2 < p < 4 \\ O\big(k^p\big(k^3 + \varepsilon'^{-2} + \varepsilon^{-2}\varepsilon'^{-1}k^2\big)\big)\log(n\kappa^{\mathsf{OL}})^{O(p)} & \text{if } p > 4 \end{cases}$$

One of our crucial insights is that the online ℓ_p Lewis weights [WY23b] (see also Section 2.2.1) can be used to bound the number of times that an optimal solution to the ℓ_p subspace approximation problem can change significantly, and furthermore, can be used to detect these changes in an online fashion. This allows us to algorithmically partition the stream into a small number of "substreams" on which a simple adaptation of sensitivity sampling is guaranteed to work, due to the fact that the optimal solution could not have changed by much.

3.3 Spectral Low Rank Approximation for Sparse Singular Vectors [WY22b]

In this section, we study algorithms for the classical problem of low rank approximation under the *spectral norm*.

Definition 3.13. *Let $\mathbf{A} \in \mathbb{R}^{n\times d}$. Then, we define the spectral norm of $\mathbf{A}$ to be*

$$\|\mathbf{A}\|_2 := \sup_{\mathbf{x}\neq 0} \frac{\|\mathbf{A}\mathbf{x}\|_2}{\|\mathbf{x}\|_2}$$

[3] Here, $\mathbf{A}_i$ is the first i rows of $\mathbf{A}$.

[4] For $a, b, \in \mathbb{R}$, we denote $\max(a,b)$ by $a \vee b$ and $\min(a,b)$ by $a \wedge b$.

Because the spectral norm is unitarily invariant, the classical Eckhart–Young–Minrky theorem [EY36, Mir60] shows that the singular value decomposition yields the optimal rank k approximation, for all k. While the singular value decomposition (SVD) can be expensive to compute for large matrices, the recent results in randomized numerical linear algebra have achieved substantial developments in fast approximation algorithms for the SVD, culminating in the following result of [MM15]:

Theorem 3.14 (Approximate spectral SVD [MM15])**.** *Let $\mathbf{A} \in \mathbb{R}^{n\times d}$. Then, there is an algorithm which computes a rank k orthogonal projection matrix $\mathbf{P} \in \mathbb{R}^{d\times d}$ such that*

$$\|\mathbf{A} - \mathbf{AP}\|_2 \leq (1+\varepsilon) \min_{\mathrm{rank}(\mathbf{A}')\leq k} \|\mathbf{A} - \mathbf{A}'\|_2$$

which runs in time at most $O(\varepsilon^{-1/2}\,\mathsf{nnz}(\mathbf{A})k\log d)$.

A natural question is whether this running time can be improved or not, under natural assumptions. One common assumption which often arises in practice is to assume that the top k singular vectors of $\mathbf{A}$ are *sparse*, i.e., there are only s nonzero values in the singular vectors. This scenario is a phenomenon known as *localization* of eigenvectors, and occurs frequently in many applications [HBCY21, ZYC+21], for example in quantum many-body problems [LVW09, NH15] and network analysis [PC18].

This question was studied in the work of [HBCY21] and a followup work of [ZYC+21], which studied algorithms for computing eigenvectors in symmetric matrices with localized eigenvectors. In [HBCY21], the authors study an algorithm for finding a small submatrix containing the supports of the leading eigenvectors by greedily adding rows and columns without formal guarantees, and [ZYC+21] seek to improve this approach using reinforcement learning techniques.

In our work of [WY22b], we obtain one of the first provable speedups over [MM15] under a sparse singular vector assumption:

Theorem 3.15 (Approximate spectral SVD for sparse singular vectors [WY22b])**.** *Let $\mathbf{A} \in \mathbb{R}^{n\times d}$ whose top k left and right singular vectors have at most s nonzero entries. Then, there is an algorithm which computes a rank k orthogonal projection matrix $\mathbf{P} \in \mathbb{R}^{d\times d}$ such that*

$$\|\mathbf{A} - \mathbf{AP}\|_2 \leq (1+\varepsilon) \min_{\mathrm{rank}(\mathbf{A}')\leq k} \|\mathbf{A} - \mathbf{A}'\|_2$$

which runs in time at most

$$O\left(\frac{\mathsf{nnz}(\mathbf{A})}{\sqrt{\varepsilon}} + \frac{n}{\varepsilon}\right)\log\frac{sdk\log n}{\varepsilon} + \mathrm{poly}\left(s, k, \frac{1}{\varepsilon}, \log n\right).$$

At a high level, our idea is to first identify a set of around $O(sk)$ (or a slightly larger number of) coordinates which contains the support of the top k singular vectors, at which point we can just output the SVD of this submatrix, padded with zeros. Thus, the difficulty lies in identifying this subset of $O(sk)$ coordinates. The work of [MM15] shows that if we know the value of the $(k+1)$th singular value $\boldsymbol{\sigma}_{k+1}$, then we can use a Chebyshev polynomial approximation of degree roughly $q = 1/\sqrt{\varepsilon}$ to identify singular vectors with singular values larger than $(1+\varepsilon)\boldsymbol{\sigma}_{k+1}$ from the vectors $\mathbf{Ag}, (\mathbf{AA}^\top)\mathbf{Ag}, \ldots, (\mathbf{AA}^\top)^q\mathbf{Ag}$, known as the *Krylov subspace*. Thus, the main problem to tackle is to find an algorithm to determine the value of $\boldsymbol{\sigma}_{k+1}$, up to a $(1+\varepsilon)$ factor. To do this, we introduce a two-stage algorithm. In the first step, we identify the value of $\boldsymbol{\sigma}_{k+1}$ up to a factor of $(1+\sqrt{\varepsilon})$ using a combination of naive power iteration together with an efficient binary searching technique over the singular values. In the second step, we know the value of $\boldsymbol{\sigma}_{k+1}$ up to a value of $(1+\sqrt{\varepsilon})$, and thus we can afford to make $1/\sqrt{\varepsilon}$ guesses to the value of $\boldsymbol{\sigma}_{k+1}$ in powers of $(1+\varepsilon)$, and add $O(sk)$ entries to our superset of the support of the sparse singular vectors for each one of the $1/\sqrt{\varepsilon}$ guesses. Then, one of these guesses will guess the right value of $\boldsymbol{\sigma}_{k+1}$, and in total, the size of our support superset is just $O(sk/\sqrt{\varepsilon})$. Our result of Theorem 3.15 follows.

References

[ABF+16] Jason M. Altschuler, Aditya Bhaskara, Gang Fu, Vahab S. Mirrokni, Afshin Rostamizadeh, and Morteza Zadimoghaddam. Greedy column subset selection: New bounds and distributed

algorithms. In Maria-Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 2539–2548. JMLR.org, 2016. 1.2

[AHV04] Pankaj K. Agarwal, Sariel Har-Peled, and Kasturi R. Varadarajan. Approximating extent measures of points. *J. ACM*, 51(4):606–635, 2004. 2.2.5

[AHV05] Pankaj K. Agarwal, Sariel Har-Peled, and Kasturi R. Varadarajan. Geometric approximation via coresets. *Combinatorial and computational geometry*, 52(1-30):3, 2005. 2.2.5

[AKPS19] Deeksha Adil, Rasmus Kyng, Richard Peng, and Sushant Sachdeva. Iterative refinement for ℓ_p-norm regression. In Timothy M. Chan, editor, *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 1405–1424. SIAM, 2019. 2.26, 1

[ALO15] Zeyuan Allen-Zhu, Zhenyu Liao, and Lorenzo Orecchia. Spectral sparsification and regret minimization beyond matrix multiplicative updates. In Rocco A. Servedio and Ronitt Rubinfeld, editors, *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 237–245. ACM, 2015. 2.2.7, 2.2.7

[APS19] Deeksha Adil, Richard Peng, and Sushant Sachdeva. Fast, provably convergent IRLS algorithm for p-norm linear regression. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 14166–14177, 2019. 2.26, 1

[AS15] Pankaj K. Agarwal and R. Sharathkumar. Streaming algorithms for extent problems in high dimensions. *Algorithmica*, 72(1):83–98, 2015. 2.2.5

[AS20] Deeksha Adil and Sushant Sachdeva. Faster p-norm minimizing flows, via smoothed q-norm problems. In Shuchi Chawla, editor, *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 892–910. SIAM, 2020. 2.26, 1

[Aue30] Herman Auerbach. *On the area of convex curves with conjugate diameters*. PhD thesis, PhD thesis, University of Lwów, 1930. 2.1.1

[BBB+19] Frank Ban, Vijay Bhattiprolu, Karl Bringmann, Pavel Kolev, Euiwoong Lee, and David P. Woodruff. A PTAS for ℓ_p-low rank approximation. In Timothy M. Chan, editor, *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 747–766. SIAM, 2019. 3.1

[BDM+20] Vladimir Braverman, Petros Drineas, Cameron Musco, Christopher Musco, Jalaj Upadhyay, David P. Woodruff, and Samson Zhou. Near optimal linear algebra in the online and sliding window models. In *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, pages 517–528. IEEE, 2020. 2.2.1, 2.2.2, 2.2.5, 3.2

[BFL16] Vladimir Braverman, Dan Feldman, and Harry Lang. New frameworks for offline and streaming coreset constructions. *CoRR*, abs/1612.00889, 2016. 2.2.2, 2.2.2

[BHM+21] Vladimir Braverman, Avinatan Hassidim, Yossi Matias, Mariano Schain, Sandeep Silwal, and Samson Zhou. Adversarial robustness of streaming algorithms through importance sampling. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 3544–3557, 2021. 2.2.2

[BJKS04] Ziv Bar-Yossef, T. S. Jayram, Ravi Kumar, and D. Sivakumar. An information statistics approach to data stream and communication complexity. *J. Comput. Syst. Sci.*, 68(4):702–732, 2004. 2.1.1

[BLM89] J. Bourgain, J. Lindenstrauss, and V. Milman. Approximation of zonoids by zonotopes. *Acta Math.*, 162(1-2):73–141, 1989. 2.2.1, 2.15, 2.2.1, 2.2.1, 2.2.1, 2.2.2, 2.2.3, 2.2.6

[BLVZ19] Aditya Bhaskara, Silvio Lattanzi, Sergei Vassilvitskii, and Morteza Zadimoghaddam. Residual based sampling for online low rank approximation. In David Zuckerman, editor, *60th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2019, Baltimore, Maryland, USA, November 9-12, 2019*, pages 1596–1614. IEEE Computer Society, 2019. 3.2

[BO10] Vladimir Braverman and Rafail Ostrovsky. Zero-one frequency laws. In Leonard J. Schulman, editor, *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010*, pages 281–290. ACM, 2010. 2.2.6

[BSS09] Joshua D. Batson, Daniel A. Spielman, and Nikhil Srivastava. Twice-ramanujan sparsifiers. In Michael Mitzenmacher, editor, *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*, pages 255–262. ACM, 2009. 2.2.7, 2.2.7

[BSS12] Joshua D. Batson, Daniel A. Spielman, and Nikhil Srivastava. Twice-ramanujan sparsifiers. *SIAM J. Comput.*, 41(6):1704–1721, 2012. 2.2.7, 2.2.7

[BW17] Christos Boutsidis and David P. Woodruff. Optimal CUR matrix decompositions. *SIAM J. Comput.*, 46(2):543–589, 2017. 3.1

[CCKW22] Nadiia Chepurko, Kenneth L. Clarkson, Praneeth Kacham, and David P. Woodruff. Near-optimal algorithms for linear algebra in the current matrix multiplication time. In Joseph (Seffi) Naor and Niv Buchbinder, editors, *Proceedings of the 2022 ACM-SIAM Symposium on Discrete Algorithms, SODA 2022, Virtual Conference / Alexandria, VA, USA, January 9 - 12, 2022*, pages 3043–3068. SIAM, 2022. 2, 2.1

[CCLY19] Michael B. Cohen, Ben Cousins, Yin Tat Lee, and Xin Yang. A near-optimal algorithm for approximating the John ellipsoid. In Alina Beygelzimer and Daniel Hsu, editors, *Conference on Learning Theory, COLT 2019, 25-28 June 2019, Phoenix, AZ, USA*, volume 99 of *Proceedings of Machine Learning Research*, pages 849–873. PMLR, 2019. 2.2.1

[CD21] Xue Chen and Michal Derezinski. Query complexity of least absolute deviation regression via robust uniform convergence. In Mikhail Belkin and Samory Kpotufe, editors, *Conference on Learning Theory, COLT 2021, 15-19 August 2021, Boulder, Colorado, USA*, volume 134 of *Proceedings of Machine Learning Research*, pages 1144–1179. PMLR, 2021. 2.2.3

[CDM+16] Kenneth L. Clarkson, Petros Drineas, Malik Magdon-Ismail, Michael W. Mahoney, Xiangrui Meng, and David P. Woodruff. The fast Cauchy transform and faster robust linear regression. *SIAM J. Comput.*, 45(3):763–810, 2016. 2.1.1

[CGK+17] Flavio Chierichetti, Sreenivas Gollapudi, Ravi Kumar, Silvio Lattanzi, Rina Panigrahy, and David P. Woodruff. Algorithms for ℓ_p low-rank approximation. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 806–814. PMLR, 2017. 3.1.1, 3.1.2

[Cla05] Kenneth L. Clarkson. Subgradient and sampling algorithms for ℓ_1 regression. In *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '05, pages 257–266, USA, 2005. Society for Industrial and Applied Mathematics. 2.1.1, 2.14, 2.2.1

[CLM+15] Michael B. Cohen, Yin Tat Lee, Cameron Musco, Christopher Musco, Richard Peng, and Aaron Sidford. Uniform sampling for matrix approximation. In Tim Roughgarden, editor, *Proceedings of the 2015 Conference on Innovations in Theoretical Computer Science, ITCS 2015, Rehovot, Israel, January 11-13, 2015*, pages 181–190. ACM, 2015. 2.2, 2.2.1

[CLS22] Cheng Chen, Yi Li, and Yiming Sun. Online active regression. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato, editors, *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 3320–3335. PMLR, 2022. 2.2.1, 2.2.3

[CMM17] Michael B. Cohen, Cameron Musco, and Christopher Musco. Input sparsity time low-rank approximation via ridge leverage score sampling. In Philip N. Klein, editor, *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, Barcelona, Spain, Hotel Porta Fira, January 16-19*, pages 1758–1777. SIAM, 2017. 3.1, 3.2

[CMP16] Michael B. Cohen, Cameron Musco, and Jakub Pachocki. Online row sampling. In Klaus Jansen, Claire Mathieu, José D. P. Rolim, and Chris Umans, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2016, September 7-9, 2016, Paris, France*, volume 60 of *LIPIcs*, pages 7:1–7:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016. 1.2, 2.2.1, 2.2.1, 2.2.5

[CMP20] Michael B. Cohen, Cameron Musco, and Jakub Pachocki. Online row sampling. *Theory Comput.*, 16:1–25, 2020. 1.2, 2.2.1, 2.2.1, 2.2.5

[Coh16] Michael B. Cohen. Nearly tight oblivious subspace embeddings by trace inequalities. In Robert Krauthgamer, editor, *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 278–287. SIAM, 2016. 2.1

[CP15] Michael B. Cohen and Richard Peng. L_p row sampling by lewis weights. In Rocco A. Servedio and Ronitt Rubinfeld, editors, *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 183–192. ACM, 2015. 2.2.1, 2.17, 2.2.1, 2.18, 2.2.1, 2.21, 2.2.1, 2.2.2

[CP19] Xue Chen and Eric Price. Active regression via linear-sample sparsification. In Alina Beygelzimer and Daniel Hsu, editors, *Conference on Learning Theory, COLT 2019, 25-28 June 2019, Phoenix, AZ, USA*, volume 99 of *Proceedings of Machine Learning Research*, pages 663–695. PMLR, 2019. 2.2.3

[CSWZ23] Yeshwanth Cherapanamjeri, Sandeep Silwal, David P. Woodruff, and Samson Zhou. Optimal algorithms for linear algebra in the current matrix multiplication time. In Nikhil Bansal and Viswanath Nagarajan, editors, *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, January 22-25, 2023*, pages 4026–4049. SIAM, 2023. 2, 2.1

[CW13] Kenneth L. Clarkson and David P. Woodruff. Low rank approximation and regression in input sparsity time. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 81–90. ACM, 2013. 2, 2.1, 3.1

[CW15a] Kenneth L. Clarkson and David P. Woodruff. Input sparsity and hardness for robust subspace approximation. In Venkatesan Guruswami, editor, *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, pages 310–329. IEEE Computer Society, 2015. 2.1.2, 2.2.2, 2.2.6, 1, 3.1.1, 3.1.1

[CW15b] Kenneth L. Clarkson and David P. Woodruff. Sketching for M-estimators: A unified approach to robust regression. In Piotr Indyk, editor, *Proceedings of the Twenty-Sixth Annual ACM-SIAM*

Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015, pages 921–939. SIAM, 2015. 3.1.1, 3.2

[CWW19] Kenneth L. Clarkson, Ruosong Wang, and David P. Woodruff. Dimensionality reduction for Tukey regression. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 1262–1271. PMLR, 2019. 2.2.6

[DDH+09] Anirban Dasgupta, Petros Drineas, Boulos Harb, Ravi Kumar, and Michael W. Mahoney. Sampling algorithms and coresets for ℓ_p regression. *SIAM J. Comput.*, 38(5):2060–2078, 2009. 2.1.1, 2.7, 2.2.1, 2.14, 2.2.1

[DG03] Sanjoy Dasgupta and Anupam Gupta. An elementary proof of a theorem of johnson and lindenstrauss. *Random Struct. Algorithms*, 22(1):60–65, 2003. 2.1.1

[DHJ+18] Chen Dan, Kristoffer Arnsfelt Hansen, He Jiang, Liwei Wang, and Yuchen Zhou. Low rank approximation of binary matrices: Column subset selection and generalizations. In Igor Potapov, Paul G. Spirakis, and James Worrell, editors, *43rd International Symposium on Mathematical Foundations of Computer Science, MFCS 2018, August 27-31, 2018, Liverpool, UK*, volume 117 of *LIPIcs*, pages 41:1–41:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018. 3.1

[DKM06a] Petros Drineas, Ravi Kannan, and Michael W. Mahoney. Fast monte carlo algorithms for matrices I: approximating matrix multiplication. *SIAM J. Comput.*, 36(1):132–157, 2006. 3, 3.1

[DKM06b] Petros Drineas, Ravi Kannan, and Michael W. Mahoney. Fast monte carlo algorithms for matrices II: computing a low-rank approximation to a matrix. *SIAM J. Comput.*, 36(1):158–183, 2006. 3, 3.1

[DKM06c] Petros Drineas, Ravi Kannan, and Michael W. Mahoney. Fast monte carlo algorithms for matrices III: computing a compressed approximate matrix decomposition. *SIAM J. Comput.*, 36(1):184–206, 2006. 3, 3.1

[DMM06a] Petros Drineas, Michael W. Mahoney, and S. Muthukrishnan. Sampling algorithms for l_2 regression and applications. In *Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2006, Miami, Florida, USA, January 22-26, 2006*, pages 1127–1136. ACM Press, 2006. 2, 2.2, 2.13, 2.2.1, 2.2.3

[DMM06b] Petros Drineas, Michael W. Mahoney, and S. Muthukrishnan. Subspace sampling and relative-error matrix approximation: Column-row-based methods. In Yossi Azar and Thomas Erlebach, editors, *Algorithms - ESA 2006, 14th Annual European Symposium, Zurich, Switzerland, September 11-13, 2006, Proceedings*, volume 4168 of *Lecture Notes in Computer Science*, pages 304–314. Springer, 2006. 3, 3.1

[DMMW12] Petros Drineas, Malik Magdon-Ismail, Michael W. Mahoney, and David P. Woodruff. Fast approximation of matrix coherence and statistical leverage. *J. Mach. Learn. Res.*, 13:3475–3506, 2012. 2, 2.2

[DTV11] Amit Deshpande, Madhur Tulsiani, and Nisheeth K. Vishnoi. Algorithms and hardness for subspace approximation. In Dana Randall, editor, *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2011, San Francisco, California, USA, Janu-ary 23-25, 2011*, pages 482–496. SIAM, 2011. 3.2

[DV06] Amit Deshpande and Santosh S. Vempala. Adaptive sampling and fast low-rank matrix approximation. In Josep Díaz, Klaus Jansen, José D. P. Rolim, and Uri Zwick, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, 9th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems, APPROX 2006 and 10th International Workshop on Randomization and Computation,*

RANDOM 2006, Barcelona, Spain, August 28-30 2006, Proceedings, volume 4110 of *Lecture Notes in Computer Science*, pages 292–303. Springer, 2006. 3, 3.1

[DV07] Amit Deshpande and Kasturi R. Varadarajan. Sampling-based dimension reduction for subspace approximation. In David S. Johnson and Uriel Feige, editors, *Proceedings of the 39th Annual ACM Symposium on Theory of Computing, San Diego, California, USA, June 11-13, 2007*, pages 641–650. ACM, 2007. 3.2

[DWZ+19] Chen Dan, Hong Wang, Hongyang Zhang, Yuchen Zhou, and Pradeep Ravikumar. Optimal analysis of subset-selection based l_p low-rank approximation. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 2537–2548, 2019. 3.1.1, 3.1.2

[EY36] Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936. 3.3

[FKV04] Alan M. Frieze, Ravi Kannan, and Santosh S. Vempala. Fast monte-carlo algorithms for finding low-rank approximations. *J. ACM*, 51(6):1025–1041, 2004. 3, 3.1

[FL11] Dan Feldman and Michael Langberg. A unified framework for approximating and clustering data. In Lance Fortnow and Salil P. Vadhan, editors, *Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011, San Jose, CA, USA, 6-8 June 2011*, pages 569–578. ACM, 2011. 2.2.2, 2.24, 2.2.2, 3.2

[FLPS22] Maryam Fazel, Yin Tat Lee, Swati Padmanabhan, and Aaron Sidford. Computing lewis weights to high precision. In Joseph (Seffi) Naor and Niv Buchbinder, editors, *Proceedings of the 2022 ACM-SIAM Symposium on Discrete Algorithms, SODA 2022, Virtual Conference / Alexandria, VA, USA, January 9 - 12, 2022*, pages 2723–2742. SIAM, 2022. 2.2.1

[FSS20] Dan Feldman, Melanie Schmidt, and Christian Sohler. Turning big data into tiny data: Constant-size coresets for k-means, pca, and projective clustering. *SIAM J. Comput.*, 49(3):601–657, 2020. 2.2.2, 2.2.2

[GPV21] Mehrdad Ghadiri, Richard Peng, and Santosh S Vempala. Faster p-norm regression using sparsity. *arXiv preprint arXiv:2109.11537*, 2021. 2.2.2, 1, 1

[GRSW12] Venkatesan Guruswami, Prasad Raghavendra, Rishi Saket, and Yi Wu. Bypassing UGC from some optimal geometric inapproximability results. In Yuval Rabani, editor, *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012, Kyoto, Japan, January 17-19, 2012*, pages 699–717. SIAM, 2012. 3.2

[GV18] Nicolas Gillis and Stephen A. Vavasis. On the complexity of robust PCA and ℓ_1-norm low-rank matrix approximation. *Math. Oper. Res.*, 43(4):1072–1084, 2018. 3.1

[HBCY21] Taylor M. Hernandez, Roel Van Beeumen, Mark A. Caprio, and Chao Yang. A greedy algorithm for computing eigenvalues of a symmetric matrix with localized eigenvectors. *Numer. Linear Algebra Appl.*, 28(2), 2021. 3.3

[HRR22] Laurel Heck, Victor Reis, and Thomas Rothvoss. The vector balancing constant for zonotopes. *CoRR*, abs/2210.16460, 2022. 1

[HV20] Lingxiao Huang and Nisheeth K. Vishnoi. Coresets for clustering in euclidean spaces: importance sampling is nearly optimal. In Konstantin Makarychev, Yury Makarychev, Madhur Tulsiani, Gautam Kamath, and Julia Chuzhoy, editors, *Proccedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020, Chicago, IL, USA, June 22-26, 2020*, pages 1416–1429. ACM, 2020. 3.2

[Ind06] Piotr Indyk. Stable distributions, pseudorandom generators, embeddings, and data stream computation. *J. ACM*, 53(3):307–323, 2006. 2.1.1, 2.3

[IW05] Piotr Indyk and David P. Woodruff. Optimal approximations of the frequency moments of data streams. In Harold N. Gabow and Ronald Fagin, editors, *Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, May 22-24, 2005*, pages 202–208. ACM, 2005. 2.1.1, 2.1.2

[JL84] William B. Johnson and Joram Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. In *Conference in modern analysis and probability (New Haven, Conn., 1982)*, volume 26 of *Contemp. Math.*, pages 189–206. Amer. Math. Soc., Providence, RI, 1984. 2.1

[JLS22] Arun Jambulapati, Yang P. Liu, and Aaron Sidford. Improved iteration complexities for overconstrained p-norm regression. In Stefano Leonardi and Anupam Gupta, editors, *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20 - 24, 2022*, pages 529–542. ACM, 2022. 2.2.1, 2.2.1

[Joh48] Fritz 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. 2.2.5

[Lee16] Yin Tat Lee. *Faster algorithms for convex and combinatorial optimization*. PhD thesis, Massachusetts Institute of Technology, 2016. 2.2.1

[Lew78] D. R. Lewis. Finite dimensional subspaces of L_p. *Studia Mathematica*, 63(2):207–212, 1978. 2.2.1, 2.15, 2.2.1, 2.17

[LLW23] Yi Li, Honghao Lin, and David P. Woodruff. The ℓ_p-subspace sketch problem in small dimensions with applications to support vector machines. In Nikhil Bansal and Viswanath Nagarajan, editors, *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, January 22-25, 2023*, pages 850–877. SIAM, 2023. 2.2.1

[LMP13] Mu Li, Gary L. Miller, and Richard Peng. Iterative row sampling. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA*, pages 127–136. IEEE Computer Society, 2013. 2.2

[LS10] Michael Langberg and Leonard J. Schulman. Universal epsilon-approximators for integrals. In Moses Charikar, editor, *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2010, Austin, Texas, USA, January 17-19, 2010*, pages 598–607. SIAM, 2010. 2.2.2, 2.24, 2.2.2, 3.2

[LS15] Yin Tat Lee and He Sun. Constructing linear-sized spectral sparsification in almost-linear time. In Venkatesan Guruswami, editor, *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, pages 250–269. IEEE Computer Society, 2015. 2.2.3, 2.2.7

[LT80] D. R. Lewis and Nicole Tomczak-Jaegermann. Hilbertian and complemented finite-dimensional subspaces of Banach lattices and unitary ideals. *J. Functional Analysis*, 35(2):165–190, 1980. 2.2.4

[LT91] Michel Ledoux and Michel Talagrand. *Probability in Banach spaces*. Classics in Mathematics. Springer-Verlag, Berlin, 1991. 2.2.1, 2.15, 2.2.1, 2.2.1, 2.2.2, 2.2.3, 2.2.6

[LVW09] Aart Lagendijk, Bart Van-Tiggelen, and Diederik S Wiersma. Fifty years of anderson localization. *Phys. Today*, 62(8):24–29, 2009. 3.3

[LWW21] Yi Li, Ruosong Wang, and David P. Woodruff. Tight bounds for the subspace sketch problem with applications. *SIAM J. Comput.*, 50(4):1287–1335, 2021. 2.2.1, 2.2.4, 2.2.7

[LWY21] Yi Li, David P. Woodruff, and Taisuke Yasuda. Exponentially improved dimensionality reduction for ℓ_1: Subspace embeddings and independence testing. In Mikhail Belkin and Samory Kpotufe, editors, *Conference on Learning Theory, COLT 2021, 15-19 August 2021, Boulder, Colorado, USA*, volume 134 of *Proceedings of Machine Learning Research*, pages 3111–3195. PMLR, 2021. (document), 1.2, 2.1.2, 2.1.2, 2.10, 2.1.3

[Mah11] Michael W. Mahoney. Randomized algorithms for matrices and data. *Found. Trends Mach. Learn.*, 3(2):123–224, 2011. 1

[Mie09] Pauli Miettinen. *Matrix decomposition methods for data mining: Computational complexity and algorithms*. PhD thesis, University of Helsinki, 2009. 3.1

[Mir60] L. Mirsky. Symmetric gauge functions and unitarily invariant norms. *Quart. J. Math. Oxford Ser. (2)*, 11:50–59, 1960. 3.3

[MM13] Xiangrui Meng and Michael W. Mahoney. Low-distortion subspace embeddings in input-sparsity time and applications to robust linear regression. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 91–100. ACM, 2013. 2.1.1, 2.1.1, 2.1.1

[MM15] Cameron Musco and Christopher Musco. Randomized block krylov methods for stronger and faster approximate singular value decomposition. In Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 1396–1404, 2015. 3.1, 3.3, 3.14, 3.3, 3.3

[MMO22] Yury Makarychev, Naren Sarayu Manoj, and Max Ovsiankin. Streaming algorithms for ellipsoidal approximation of convex polytopes. In Po-Ling Loh and Maxim Raginsky, editors, *Conference on Learning Theory, 2-5 July 2022, London, UK*, volume 178 of *Proceedings of Machine Learning Research*, pages 3070–3093. PMLR, 2022. 2.2.5

[MMR21] Tung Mai, Cameron Musco, and Anup Rao. Coresets for classification - simplified and strengthened. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 11643–11654, 2021. 2.2.6

[MMWY22] Cameron Musco, Christopher Musco, David P. Woodruff, and Taisuke Yasuda. Active linear regression for ℓ_p norms and beyond. In *63rd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2022, Denver, CO, USA, October 31 - November 3, 2022*, pages 744–753. IEEE, 2022. (document), 1.2, 2.2.1, 2.2.3, 2.2.3, 2.32, 2.2.6, 2.2.6, 2.38, 2.2.6, 2.39, 1, 3.1.1

[MOW23] Alexander Munteanu, Simon Omlor, and David Woodruff. Almost linear constant-factor sketching for ℓ_1 and logistic regression. In *The Eleventh International Conference on Learning Representations*, 2023. 2.1.2

[MSS10] Asish Mukhopadhyay, Animesh Sarker, and Tom Switzer. Approximate ellipsoid in the streaming model. In Weili Wu and Ovidiu Daescu, editors, *Combinatorial Optimization and Applications - 4th International Conference, COCOA 2010, Kailua-Kona, HI, USA, December 18-20, 2010, Proceedings, Part II*, volume 6509 of *Lecture Notes in Computer Science*, pages 401–413. Springer, 2010. 2.2.5

[MSSW18] Alexander Munteanu, Chris Schwiegelshohn, Christian Sohler, and David P. Woodruff. On coresets for logistic regression. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 6562–6571, 2018. 2.2.2, 2.2.6

[MT20] Per-Gunnar Martinsson and Joel A. Tropp. Randomized numerical linear algebra: Foundations and algorithms. *Acta Numer.*, 29:403–572, 2020. 1

[MW21] Arvind V. Mahankali and David P. Woodruff. Optimal ℓ_1 column subset selection and a fast PTAS for low rank approximation. In Dániel Marx, editor, *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA 2021, Virtual Conference, January 10 - 13, 2021*, pages 560–578. SIAM, 2021. 3.1, 3.1.1, 3.1.2, 3.1.2

[NH15] Rahul Nandkishore and David A Huse. Many-body localization and thermalization in quantum statistical mechanics. *Annu. Rev. Condens. Matter Phys.*, 6(1):15–38, 2015. 3.3

[NN13] Jelani Nelson and Huy L. Nguyen. OSNAP: faster numerical linear algebra algorithms via sparser subspace embeddings. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA*, pages 117–126. IEEE Computer Society, 2013. 2.1

[Nol20] John P. Nolan. *Univariate stable distributions: models for heavy tailed data.* Springer Series in Operations Research and Financial Engineering. Springer, Cham, [2020] ©2020. 2.3, 3.1.2

[PC18] Romualdo Pastor-Satorras and Claudio Castellano. Eigenvector localization in real networks and its implications for epidemic spreading. *Journal of Statistical Physics*, 173(3):1110–1123, 2018. 3.3

[PPP21] Aditya Parulekar, Advait Parulekar, and Eric Price. L1 regression with Lewis weights subsampling. In Mary Wootters and Laura Sanità, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2021, August 16-18, 2021, University of Washington, Seattle, Washington, USA (Virtual Conference)*, volume 207 of *LIPIcs*, pages 49:1–49:21. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. 2.2.3

[RV07] Mark Rudelson and Roman Vershynin. Sampling from large matrices: An approach through geometric functional analysis. *J. ACM*, 54(4):21, 2007. 2.13, 2.2.1

[Sar06] Tamás Sarlós. Improved approximation algorithms for large matrices via random projections. In *47th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2006), 21-24 October 2006, Berkeley, California, USA, Proceedings*, pages 143–152. IEEE Computer Society, 2006. 2, 2.1, 2.2, 2.1, 2.2.3

[Sch07] Gideon Schechtman. Aimpl: Fourier analytic methods in convex geometry, available at http://aimpl.org/fourierconvex/1/, 2007. 1

[Sch11] Gideon Schechtman. Tight embedding of subspaces of L_p in ℓ_p^n for even p. *Proc. Amer. Math. Soc.*, 139(12):4419–4421, 2011. 1

[SS02] Michael E. Saks and Xiaodong Sun. Space lower bounds for distance approximation in the data stream model. In John H. Reif, editor, *Proceedings on 34th Annual ACM Symposium on Theory of Computing, May 19-21, 2002, Montréal, Québec, Canada*, pages 360–369. ACM, 2002. 2.1.1

[SS11] Daniel A. Spielman and Nikhil Srivastava. Graph sparsification by effective resistances. *SIAM J. Comput.*, 40(6):1913–1926, 2011. 2.2

[SW11] Christian Sohler and David P. Woodruff. Subspace embeddings for the l_1-norm with applications. In Lance Fortnow and Salil P. Vadhan, editors, *Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011, San Jose, CA, USA, 6-8 June 2011*, pages 755–764. ACM, 2011. 2.1.1, 2.1.1, 2.1.2, 2.1.2

[SWZ17] Zhao Song, David P. Woodruff, and Peilin Zhong. Low rank approximation with entrywise ℓ_1-norm error. In Hamed Hatami, Pierre McKenzie, and Valerie King, editors, *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 688–701. ACM, 2017. 3.1.1, 3.1.2, 3.1.2

[SWZ19] Zhao Song, David P. Woodruff, and Peilin Zhong. Towards a zero-one law for column subset selection. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 6120–6131, 2019. 3.1.1, 3.1.1, 3.1.1

[SZ01] Gideon Schechtman and Artem Zvavitch. Embedding subspaces of l_p into l_p^n, $0 < p < 1$. *Mathematische Nachrichten*, 227(1):133–142, 2001. 2.2.1, 2.15, 2.2.1, 2.2.1, 2.2.3, 2.2.6

[Tal90] Michel Talagrand. Embedding subspaces of L_1 into l_1^N. *Proc. Amer. Math. Soc.*, 108(2):363–369, 1990. 2.2.1

[Tal95] Michel Talagrand. Embedding subspaces of L_p in l_p^N. In *Geometric aspects of functional analysis (Israel, 1992–1994)*, volume 77 of *Oper. Theory Adv. Appl.*, pages 311–325. Birkhäuser, Basel, 1995. 2.2.1

[TMF20] Murad Tukan, Alaa Maalouf, and Dan Feldman. Coresets for near-convex functions. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. 2.2.6

[Tod16] Michael J. Todd. *Minimum volume ellipsoids - theory and algorithms*, volume 23 of *MOS-SIAM Series on Optimization*. SIAM, 2016. 2.1.1, 2.2.5

[VX12] Kasturi R. Varadarajan and Xin Xiao. On the sensitivity of shape fitting problems. In Deepak D'Souza, Telikepalli Kavitha, and Jaikumar Radhakrishnan, editors, *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2012, December 15-17, 2012, Hyderabad, India*, volume 18 of *LIPIcs*, pages 486–497. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2012. 3.2

[Woo14] David P. Woodruff. Sketching as a tool for numerical linear algebra. *Found. Trends Theor. Comput. Sci.*, 10(1-2):1–157, 2014. 1

[WW19] Ruosong Wang and David P. Woodruff. Tight bounds for ℓ_p oblivious subspace embeddings. In Timothy M. Chan, editor, *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 1825–1843. SIAM, 2019. 2.1.1, 2.4, 2.1.1, 2.5, 2.1.1, 2.1.1, 2.1.2, 2.9, 2.1.2

[WW22] Ruosong Wang and David P. Woodruff. Tight bounds for ℓ_1 oblivious subspace embeddings. *ACM Trans. Algorithms*, 18(1):8:1–8:32, 2022. 2.1.1, 2.4, 2.1.1, 2.5, 2.1.1, 2.1.2, 2.9, 2.1.2

[WY22a] David P. Woodruff and Taisuke Yasuda. High-dimensional geometric streaming in polynomial space. In *63rd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2022, Denver, CO, USA, October 31 - November 3, 2022*, pages 732–743. IEEE, 2022. (document), 1.2, 1.2, 2.2.1, 2.2.2, 2.2.4, 2.2.4, 2.34, 2.35, 2.2.5, 2.2.5, 2.37

[WY22b] David P. Woodruff and Taisuke Yasuda. Improved algorithms for low rank approximation from sparsity. In Joseph (Seffi) Naor and Niv Buchbinder, editors, *Proceedings of the 2022 ACM-SIAM Symposium on Discrete Algorithms, SODA 2022, Virtual Conference / Alexandria, VA, USA, January 9 - 12, 2022*, pages 2358–2403. SIAM, 2022. (document), 1.2, 3.3, 3.3, 3.15

[WY23a] David P. Woodruff and Taisuke Yasuda. New subset selection algorithms for low rank approximation: Offline and online. In *Symposium on Theory of Computing Conference, STOC'23*. ACM, 2023. (document), 1.2, 1.2, 2.1.1, 2.1.1, 2.6, 2.1.1, 2.8, 2.2.3, 2.2.3, 2.32, 3.1, 3.8, 3.2, 3.2

[WY23b] David P. Woodruff and Taisuke Yasuda. Online lewis weight sampling. In Nikhil Bansal and Viswanath Nagarajan, editors, *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, January 22-25, 2023*, pages 4622–4666. SIAM, 2023. (document), 1.2, 2.2.1, 2.2.1, 2.2.1, 2.20, 2.2.1, 2.2.1, 2.23, 3.2

[WY23c] David P. Woodruff and Taisuke Yasuda. Sharper bounds for ℓ_p sensitivity sampling. *Preprint*, 2023. (document), 2.2.2, 2.2.2, 2.28, 2.29, 1, 1

[WZ13] David P. Woodruff and Qin Zhang. Subspace embeddings and ℓ_p-regression using exponential random variables. In Shai Shalev-Shwartz and Ingo Steinwart, editors, *COLT 2013 - The 26th Annual Conference on Learning Theory, June 12-14, 2013, Princeton University, NJ, USA*, volume 30 of *JMLR Workshop and Conference Proceedings*, pages 546–567. JMLR.org, 2013. 2.1.1, 2.1.1, 2.1.1

[Zou12] Anastasios Zouzias. A matrix hyperbolic cosine algorithm and applications. In Artur Czumaj, Kurt Mehlhorn, Andrew M. Pitts, and Roger Wattenhofer, editors, *Automata, Languages, and Programming - 39th International Colloquium, ICALP 2012, Warwick, UK, July 9-13, 2012, Proceedings, Part I*, volume 7391 of *Lecture Notes in Computer Science*, pages 846–858. Springer, 2012. 2.2.7, 2.2.7

[Zva00] A. Zvavitch. More on embedding subspaces of L_p into l_p^N, $0 < p < 1$. In *Geometric aspects of functional analysis*, volume 1745 of *Lecture Notes in Math.*, pages 269–280. Springer, Berlin, 2000. 2.2.1

[ZYC+21] Li Zhou, Lihao Yan, Mark A. Caprio, Weiguo Gao, and Chao Yang. Solving the k-sparse eigenvalue problem with reinforcement learning. *CSIAM Transactions on Applied Mathematics*, 2(4):697–723, 2021. 3.3