SIAM J. COMPUT.
Vol. 45, No. 3, pp. 763–810

THE FAST CAUCHY TRANSFORM AND FASTER ROBUST LINEAR REGRESSION*

KENNETH L. CLARKSON†, PETROS DRINEAS‡, MALIK MAGDON-ISMAIL‡, MICHAEL W. MAHONEY§, XIANGRUI MENG¶, AND DAVID P. WOODRUFF†

Abstract. We provide fast algorithms for overconstrained ℓ_p regression and related problems: for an $n \times d$ input matrix A and vector $b \in \mathbb{R}^n$, in $O(nd \log n)$ time we reduce the problem $\min_{x \in \mathbb{R}^d} \|Ax - b\|_p$ to the same problem with input matrix $\tilde{A}$ of dimension $s \times d$ and corresponding $\tilde{b}$ of dimension $s \times 1$. Here, $\tilde{A}$ and $\tilde{b}$ are a *coreset* for the problem, consisting of sampled and rescaled rows of A and b; and s is independent of n and polynomial in d. Our results improve on the best previous algorithms when $n \gg d$ for all $p \in [1, \infty)$ except $p = 2$; in particular, they improve the $O(nd^{1.376+})$ running time of Sohler and Woodruff [*Proceedings of the 43rd Annual ACM Symposium on Theory of Computing,* 2011, pp. 755–764] for $p = 1$, which uses asymptotically fast matrix multiplication, and the $O(nd^5 \log n)$ time of Dasgupta et al. [*SIAM J. Comput.*, 38 (2009), pp. 2060–2078] for general p, which uses ellipsoidal rounding. We also provide a suite of improved results for finding well-conditioned bases via ellipsoidal rounding, illustrating tradeoffs between running time and conditioning quality, including a one-pass conditioning algorithm for general ℓ_p problems. To complement this theory, we provide a detailed empirical evaluation of implementations of our algorithms for $p = 1$, comparing them with several related algorithms. Among other things, our empirical results clearly show that, in the asymptotic regime, the theory is a very good guide to the practical performance of these algorithms. Our algorithms use our faster constructions of well-conditioned bases for ℓ_p spaces and, for $p = 1$, a fast subspace embedding of independent interest that we call the Fast Cauchy transform: a distribution over matrices $\Pi : \mathbb{R}^n \mapsto \mathbb{R}^{O(d \log d)}$, found obliviously to A, that approximately preserves the ℓ_1 norms, that is, with large probability, simultaneously for all x, $\|Ax\|_1 \approx \|\Pi Ax\|_1$, with distortion $O(d^{2+\eta})$, for an arbitrarily small constant $\eta > 0$; and, moreover, ΠA can be computed in $O(nd \log d)$ time. The techniques underlying our Fast Cauchy transform include Fast Johnson–Lindenstrauss transforms, low-coherence matrices, and rescaling by Cauchy random variables.

Key words. randomized algorithms, subspace embedding, robust regression, Cauchy transform

AMS subject classification. 68W20

DOI. 10.1137/140963698

1. Introduction.

Random sampling, random projection, and other embedding methods have proven to be very useful in recent years in the development of improved worst-case algorithms for a range of linear algebra problems. For example, Gaussian random projections provide low-distortion subspace embeddings in the ℓ_2 norm, mapping an arbitrary d-dimensional subspace in $\mathbb{R}^n$ into a d-dimensional subspace in $\mathbb{R}^r$, with $r = O(d)$, and distorting the ℓ_2 norm of each vector in the subspace by at most a constant factor. Importantly for many applications, the embedding is oblivious in

*Received by the editors April 7, 2014; accepted for publication (in revised form) March 11, 2016; published electronically June 15, 2016. A conference version of this paper appeared under the same title in *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*, 2013.

http://www.siam.org/journals/sicomp/45-3/96369.html

†IBM Almaden Research Center, San Jose, CA 95120 (klclarks@us.ibm.com, dpwoodru@us.ibm.com).

‡Department of Computer Science, Rensselaer Polytechnic Institute, Troy, NY 12180 (drinep@cs.rpi.edu, magdon@cs.rpi.edu). The second author's research was supported by NSF grants IIS-1447283 and IIS-1302231.

§International Computer Science Institute and Department of Statistics, University of California at Berkeley, Berkeley, CA 94720 (mmahoney@stat.berkeley.edu). This author's research was partially supported by grants from the NSF, ARO, and DARPA.

¶Databricks, San Francisco, CA 94105 (meng@databricks.com).

the sense that it is implemented by a linear mapping chosen from a distribution on mappings that is independent of the input subspace. Such low-distortion embeddings can be used to speed up various geometric algorithms if they can be computed sufficiently quickly. As an example, the *Fast Johnson–Lindenstrauss transform* (FJLT) is one such embedding; the FJLT is computable in $O(n \log d)$ time, using a variant of the Fast Hadamard transform [1]. Among other things, use of the FJLT leads to faster algorithms for constructing orthonormal bases, ℓ_2 regression, and ℓ_2 subspace approximation, which in turn lead to faster algorithms for a range of related problems including low-rank matrix approximation [12, 20, 10].

In this paper, we use ℓ_1 and ℓ_p extensions of these methods to provide faster algorithms for the classical ℓ_p regression problem and several other related problems. Recall the overconstrained ℓ_p regression problem.

DEFINITION 1. *Given a matrix $A \in \mathbb{R}^{n \times d}$, with $n > d$, a vector $b \in \mathbb{R}^n$, and a norm $\|\cdot\|_p$, the* ℓ_p regression problem *is to find an optimal solution to*

$$\min_{x \in \mathbb{R}^d} \|Ax - b\|_p. \tag{1}$$

In this paper, we are most interested in the case $p = 1$, although many of our results hold more generally, and so we state several of our results for general p. The ℓ_1 regression problem, also known as the least absolute deviations or least absolute errors problem, is especially of interest as a more robust alternative to the ℓ_2 regression or least squares approximation problem.

It is well known that for $p \geq 1$, the ℓ_p regression problem is a convex optimization problem; and for $p = 1$ and $p = \infty$, it is an instance of linear programming. Recent work has focused on using sampling, projection, and other embedding methods to solve these problems more quickly than with general convex programming or linear programming methods. Most relevant for our work is the work of Clarkson [6] on solving the ℓ_1 regression problem with subgradient and sampling methods; the work of Dasgupta et al. [7] on using well-conditioned bases and subspace-preserving sampling algorithms to solve general ℓ_p regression problems; and the work of Sohler and Woodruff [24] on using the Cauchy transform to obtain improved ℓ_1 embeddings, thereby leading to improved algorithms for the ℓ_1 regression problem. The Cauchy transform of [24] provides low-distortion embeddings for the ℓ_1 norm, and thus it is an ℓ_1 analogue of the Gaussian projection for ℓ_2. It consists of a dense matrix of Cauchy random variables, and so it is "slow" to apply to an arbitrary matrix A; but since it provides the first analogue of the Johnson–Lindenstrauss embedding for the ℓ_1 norm, it can be used to speed up randomized algorithms for problems such as ℓ_1 regression and ℓ_1 subspace approximation [24].

In this paper, we provide fast algorithms for overconstrained ℓ_p regression and several related problems. Our algorithms use our faster constructions of well-conditioned bases for ℓ_p spaces; and, for $p = 1$, our algorithms use a fast subspace embedding of independent interest that we call the Fast Cauchy transform (FCT). We also provide a detailed empirical evaluation of the FCT and its use at computing ℓ_1 well-conditioned bases and solving ℓ_1 regression problems.

The FCT is our main technical result, and it is essentially an ℓ_1 analogue of the FJLT. The FCT can be represented by a distribution over matrices $\Pi : \mathbb{R}^n \mapsto \mathbb{R}^{O(d \log d)}$, found obliviously to A (in the sense that its construction does not depend on any information in A), that approximately preserves the ℓ_1 norms of all vectors in $\{Ax \mid x \in \mathbb{R}^d\}$. That is, with large probability, simultaneously for all x, $\|Ax\|_1 \approx \|\Pi Ax\|_1$, with distortion $O(d^{2+\eta} \log d)$, for an arbitrarily small constant $\eta > 0$ (see

Theorem 2); and, moreover, ΠA can be computed in $O(nd \log d)$ time. We actually provide two related constructions of the FCT (see Theorems 1 and 2). The techniques underlying our FCTs include FJLTs, low-coherence matrices, and rescaling by Cauchy random variables.

Our main application of the FCT embedding is to constructing the current fastest algorithm for computing a *well-conditioned basis* for ℓ_1 (see Theorem 4). Such a basis is an analogue for the ℓ_1 norm of what an orthonormal basis is for the ℓ_2 norm, and our result improves the result in [24]. We also provide a generalization of this result to constructing ℓ_p well-conditioned bases (see Theorem 11). The main application for well-conditioned bases is to regression: if the rows of A are sampled according to probabilities derived from the norms of the rows of such a basis, the resulting sample of rows (and corresponding entries of b) are with high probability a *coreset* for the regression problem; see, e.g., [7]. That is, for an $n \times d$ input matrix A and vector $b \in \mathbb{R}^n$, we can reduce an ℓ_p regression problem to another ℓ_p regression problem with input matrix $\tilde{A}$ of dimension $s \times d$ and corresponding $\tilde{b}$ of dimension $s \times 1$. Here, $\tilde{A}$ and $\tilde{b}$ consist of sampled and rescaled rows of A and b; and s is independent of n and polynomial in d. We point out that our construction uses as a black box an FJLT, which means that any improvement in the running time of the FJLT (for example, exploiting the sparsity of A) results in a corresponding improvement to the running times of our ℓ_p regression.

Based on our constructions of well-conditioned bases, we give the fastest known construction of coresets for ℓ_p regression for all $p \in [1, \infty)$ except $p = 2$. In particular, for ℓ_1 regression, we construct a coreset of size $\frac{1}{\varepsilon^2} \operatorname{poly}(d, \log \frac{1}{\varepsilon})$ that achieves a $(1+\varepsilon)$-approximation guarantee (see Theorem 5). Our construction runs in $O(nd \log n)$ time, improving the previous best algorithm of Sohler and Woodruff [24], which has an $O(nd^{1.376+})$ running time. Our extension to finding an ℓ_p well-conditioned basis also leads to an $O(nd \log n)$ time algorithm for a $(1+\varepsilon)$-approximation to the ℓ_p regression problem (see Theorem 12), improving the $O(nd^5 \log n)$ algorithm of Dasgupta et al. [7]. For $p = 1$, extensions of our basic methods yield improved algorithms for several related problems. For example, we actually further optimize the running time for $p = 1$ to $O(nd \log(\varepsilon^{-1} d \log n))$ (see Theorem 6). In addition, we generalize our ℓ_1 result to solving the multiple regression problem (see Theorem 7); and we use this to give the current fastest algorithm for computing a $(1 + \varepsilon)$-approximation for the ℓ_1 subspace approximation problem (see Theorem 8).

In addition to our construction of ℓ_p well-conditioned bases (see Theorem 11) and their use in providing a $(1 + \varepsilon)$-approximation to the ℓ_p regression problem (see Theorem 12), we also provide a suite of improved results for finding well-conditioned bases via ellipsoidal rounding for general ℓ_p problems, illustrating tradeoffs between running time and conditioning quality. These methods complement the FCT-based methods in the sense that the FCT may be viewed as a tool to compute a good basis in an oblivious manner, and the ellipsoid-based methods provide an alternate way to compute a good basis in a data-dependent manner. In particular, we prove that we can obtain an ellipsoidal rounding matrix in at most $O(nd^3 \log n)$ time that provides a $2d$-rounding (see Theorem 10). This is much faster than the algorithm of Lovász [19] that computes a $(d(d + 1))^{1/2}$-rounding in $O(nd^5 \log n)$ time. We also present an optimized algorithm that uses an FJLT to compute a well-conditioned basis of A in $O(nd \log n)$ time (see Theorem 11). When $p = 1$, these ℓ_p rounding algorithms are competitive with or better than previous algorithms that were developed for ℓ_1.

Finally, we also provide the first empirical evaluation for this class of randomized

algorithms. In particular, we provide a detailed evaluation of a numerical implementation of both FCT constructions, and we compare the results with an implementation of the (slow) Cauchy transform, as well as a Gaussian transform and an FJLT. These latter two are ℓ_2-based projections. We evaluate the quality of the ℓ_1 well-conditioned basis, the core component in all our geometric algorithms, on a suite of matrices designed to test the limits of these randomized algorithms, and we also evaluate how the method performs in the context of ℓ_1 regression. This latter evaluation includes an implementation on a nearly terabyte-scale problem, where we achieve a 10^{-3} relative-error approximation to the optimal solution, a task that was infeasible prior to our work. Among other things, our empirical results clearly show that, in the asymptotic regime, the theory is a very good guide to the practical performance of these algorithms.

Since this paper is long and detailed, we provide here a brief outline. We start in section 2 with some preliminaries, including several technical results that we will use in our analysis and that are of independent interest. Then, in section 3, we will present our main technical results for the FCT; and in section 4, we will describe applications of it to ℓ_1 well-conditioned basis construction and ℓ_1 leverage score approximation, to solving the ℓ_1 regression problem, and to solving the ℓ_1 norm subspace approximation problem. Then, in section 5, we describe extensions of these ideas to general ℓ_p problems. Section 6 will contain a detailed empirical evaluation of our algorithms for ℓ_1-based problems, including the construction of ℓ_1 well-conditioned bases and both small-scale and large-scale ℓ_1 regression problems. Section 7 will then contain a brief conclusion. For simplicity of presentation, the proofs of many of our results have been moved to Appendices A through H.

2. Preliminaries. Let $A \in \mathbb{R}^{n \times d}$ be an $n \times d$ input matrix, where we assume $n \gg d$ and A has full column rank. The task of *linear regression* is to find a vector $x^* \in \mathbb{R}^d$ that minimizes $\|Ax - b\|$ with respect to x for a given $b \in \mathbb{R}^n$ and norm $\|\cdot\|$. In this paper, our focus is mostly on the ℓ_1 norm, although we also discuss extensions to ℓ_p for any $p \geq 1$. Recall that, for $p \in [1, \infty]$, the ℓ_p norm of a vector x is $\|x\|_p = (\sum_i |x_i|^p)^{1/p}$, defined to be $\max_i |x_i|$ for $p = \infty$. Let $[n]$ denote the set $\{1, 2, \ldots, n\}$; and let $A_{(i)}$ and $A^{(j)}$ be the ith row vector and jth column vector of A, respectively. For matrices, we use the Frobenius norm $\|A\|_F^2 = \sum_{i=1}^n \sum_{j=1}^d A_{ij}^2$, the ℓ_2-operator (or spectral) norm $\|A\|_2 = \sup_{\|x\|_2=1} \|Ax\|_2$, and the entrywise ℓ_p norm $\|X\|_p = (\sum_{i,j} |X_{ij}|^p)^{1/p}$. (The exception to this is $p = 2$, where this notation is used for the spectral norm and the entrywise 2-norm is the Frobenius norm.) Finally, the standard inner product between vectors x, y is $\langle x, y \rangle = x^T y$; e_i are standard basis vectors of the relevant dimension; I_n denotes the $n \times n$ identity matrix; and c refers to a generic constant whose specific value may vary throughout the paper.

Two useful tail inequalities. The following two Bernstein-type tail inequalities are useful because they give tail bounds without reference to the number of independent and identically distributed (i.i.d.) trials. The first bound is due to Maurer [22], and the second is an immediate application of the first.

LEMMA 1 (see [22]). *Let $X_i \geq 0$ be independent random variables with $\sum_i \mathbf{E}[X_i^2] < \infty$, and define $X = \sum_i X_i$. Then, for any $t > 0$,*

$$\mathbf{Pr}[X \leq \mathbf{E}[X] - t] \leq \exp\left(\frac{-t^2}{2\sum_i \mathbf{E}[X_i^2]}\right).$$

LEMMA 2. *Let x_i be i.i.d. Bernoulli random variables with probability p, and let $X = \sum_{i\in[n]} \xi_i x_i$, where $\xi_i \geq 0$, with $\sum_{i\in[n]} \xi_i = \xi$ and $\sum_{i\in[n]} \xi_i^2 \leq \xi^2/\beta^2$. Then, for any $t > 0$,*

$$\mathbf{Pr}[X \geq \xi(p+t)] \leq \exp\left(-\frac{\beta^2 t^2}{2(1-p)}\right).$$

Proof. The proof is a straightforward application of Lemma 1 to $Z = \sum_{i\in[n]} \xi_i(1-x_i)$. ∎

Sums of Cauchy random variables. The Cauchy distribution, having density $p(x) = \frac{1}{\pi}\frac{1}{1+x^2}$, is the unique 1-stable distribution. If $C_1, \ldots, C_M$ are independent Cauchys, then $\sum_{i\in[M]} \gamma_i C_i$ is distributed as a Cauchy scaled by $\gamma = \sum_{i\in[M]} |\gamma_i|$. The Cauchy distribution will factor heavily in our discussion, and bounds for sums of Cauchy random variables will be used throughout. We note that the Cauchy distribution has undefined expectation and infinite variance.

The following upper and lower tail inequalities for sums of Cauchy random variables are proved in Appendix A. The proof of Lemma 3 is similar to an argument of Indyk [15], though in that paper the Cauchy random variables are independent. As in that paper, our argument follows by a Markov bound after conditioning on the magnitudes of the Cauchy random variable summands not being too large, so that their conditional expectations are defined. However, in this paper, the Cauchy random variables are dependent, and so after conditioning on a global event, the expectations of the magnitudes need not be the same as after this conditioning in the independent case. Lemma 4 is a simple application of Lemma 1, while Lemma 5 was shown in [7]; we include the proofs for completeness.

LEMMA 3 (Cauchy upper tail inequality). *For $i \in [m]$, let C_i be m (not necessarily independent) Cauchy random variables, and $\gamma_i > 0$ with $\gamma = \sum_{i\in[m]} \gamma_i$. Let $X = \sum_{i\in[m]} \gamma_i |C_i|$. Then, for any $t \geq 1$,*

$$\mathbf{Pr}\left[X > \gamma t\right] \leq \frac{1}{\pi t}\left(\frac{\log(1+(2mt)^2)}{1-1/(\pi t)} + 1\right) = \frac{\log(mt)}{t}\left(1+o(1)\right).$$

Remark. The bound has only logarithmic dependence on the number of Cauchy random variables and does *not* rely on any independence assumption among the random variables. Even if the Cauchys are independent, one cannot substantially improve on this bound due to the nature of the Cauchy distribution. This is because, for independent Cauchys, $\sum_i \gamma_i |C_i| \geq |\sum_i \gamma_i C_i|$, and the latter sum is itself distributed as a Cauchy scaled by γ. Hence for independent Cauchys, $\mathbf{Pr}[X \geq \gamma t] = \frac{2}{\pi}\tan^{-1} t = \Omega(\frac{1}{t})$.

LEMMA 4 (Cauchy lower tail inequality). *For $i \in [r]$, let C_i be independent Cauchy random variables, and $\gamma_i \geq 0$ with $\gamma = \sum_{i\in[r]} \gamma_i$ and $\sum_{i\in[r]} \gamma_i^2 \leq \gamma^2/\beta^2$. Let $X = \sum_{i\in[r]} \gamma_i |C_i|$. Then, for any $t \geq 0$,*

$$\mathbf{Pr}\left[X \leq \gamma(1-t)\right] \leq \exp\left(-\frac{\beta^2 t^2}{3}\right).$$

An ℓ_1 sampling lemma. We will also need an "ℓ_1-sampling lemma," which is an application of Bernstein's inequality. This lemma bounds how ℓ_1 norms get distorted under sampling according to ℓ_1 probabilities. The proof of this lemma is also given in Appendix A.

LEMMA 5 (ℓ_1 sampling lemma). *Let $Z \in \mathbb{R}^{n\times k}$, and suppose that for $i \in [n]$, $t_i \geq a\|Z_{(i)}\|_1/\|Z\|_1$. For $s > 0$, define $\hat{p}_i = \min\{1, s \cdot t_i\}$, and let $D \in \mathbb{R}^{n\times n}$ be a*

random diagonal matrix with $D_{ii} = 1/\hat{p}_i$ *with probability* $\hat{p}_i$*, and* 0 *otherwise. Then, for any (fixed)* $x \in \mathbb{R}^k$*, with probability at least* $1 - \delta$*,*

$$(1-\varepsilon)\|Zx\|_1 \le \|DZx\|_1 \le (1+\varepsilon)\|Zx\|_1,$$

where $\delta \le 2\exp\left(\frac{-as\varepsilon^2\|Zx\|_1}{(2+\frac{2}{3}\varepsilon)\|Z\|_1\|x\|_\infty}\right)$.

3. Main technical result: The Fast Cauchy transform. In this section, we present the Fast Cauchy transform (FCT), which is an ℓ_1-based analogue of the Fast Johnson–Lindenstrauss transform (FJLT). We will actually present two related constructions, one based on using a quickly constructable low-coherence matrix and one based on using a version of the FJLT. In both cases, these matrices will be rescaled by Cauchy random variables (hence the name *Fast Cauchy transform*). We will also state our main results, Theorems 1 and 2, which provide running times and quality-of-approximation guarantees for these two FCT embeddings.

3.1. FCT1 construction: Via a low-coherence matrix. This FCT construction first preprocesses by a deterministic low-coherence "spreading matrix," then rescales by Cauchy random variables, and finally samples linear combinations of the rows. Let $\delta \in (0, 1]$ be a parameter governing the failure probability of our algorithm. Then, we construct Π_1 as

$$\Pi_1 \equiv 4BC\tilde{H},$$

where the following hold:

- $B \in \mathbb{R}^{r_1 \times 2n}$ has each column chosen independently and uniformly from the r_1 standard basis vectors for $\mathbb{R}^{r_1}$; we will set the parameter $r_1 = \alpha d \log\frac{d}{\delta}$, where δ controls the probability that our algorithms fail and α is a suitably large constant.
- $C \in \mathbb{R}^{2n\times 2n}$ is a diagonal matrix with diagonal entries chosen independently from a Cauchy distribution.
- $\tilde{H} \in \mathbb{R}^{2n\times n}$ is a block-diagonal matrix comprised of n/s blocks along the diagonal. Each block is the $2s \times s$ matrix $G_s \equiv \left[\begin{smallmatrix} H_s \\ I_s \end{smallmatrix}\right]$, where I_s is the $s \times s$ identity matrix, and H_s is the normalized Hadamard matrix. We will set $s = r_1^6$. (Here, for simplicity, we assume s is a power of two and n/s is an integer.)

$$\tilde{H} \equiv \begin{bmatrix} G_s & & & \\ & G_s & & \\ & & \ddots & \\ & & & G_s \end{bmatrix}.$$

(For completeness, we remind the reader that the (nonnormalized) $n \times n$ matrix of the Hadamard transform H_n may be defined recursively as follows:

$$H_n = \begin{bmatrix} H_{n/2} & H_{n/2} \\ H_{n/2} & -H_{n/2} \end{bmatrix}, \quad \text{with} \quad H_2 = \begin{bmatrix} +1 & +1 \\ +1 & -1 \end{bmatrix}.$$

The $n \times n$ normalized matrix of the Hadamard transform is then equal to $\frac{1}{\sqrt{n}}H_n$; hereafter, we will denote this normalized matrix by H_n.)

Remark. Heuristically, the effect of $\tilde{H}$ in the above FCT construction is to spread the weight of a vector, so that $\tilde{H}y$ has many entries that are not too small. (This is discussed in Lemma 6 in the proof of Theorem 1 below.) This means that the vector

$C\tilde{H}y$ comprises Cauchy random variables with scale factors that are not too small; and finally these variables are summed up by B, yielding a vector $BC\tilde{H}y$, whose ℓ_1 norm will not be too small relative to $\|y\|_1$.

For this version of the FCT, we have the following theorem. The proof of this theorem may be found in section 3.3.

THEOREM 1 (Fast Cauchy transform (FCT1)). *There is a distribution (given by the above construction) over matrices $\Pi_1 \in \mathbb{R}^{r_1 \times n}$, with $r_1 = O(d \log d + d \log \frac{1}{\delta})$, such that for an arbitrary (but fixed) $A \in \mathbb{R}^{n \times d}$, and for all $x \in \mathbb{R}^d$, the inequalities*

$$\|Ax\|_1 \leq \|\Pi_1 Ax\|_1 \leq \kappa \|Ax\|_1 \tag{2}$$

hold with probability $1 - \delta$, where

$$\kappa = O\left(\frac{d\sqrt{s}}{\delta} \log(r_1 d)\right).$$

Further, for any $y \in \mathbb{R}^n$, the product $\Pi_1 y$ can be computed in $O(n \log r_1)$ time.

Setting δ to a small constant, since $\sqrt{s} = r_1^3$ and $r_1 = O(d \log d)$, it follows that $\kappa = O(d^4 \log^4 d)$ in the above theorem.

Remark. The existence of such a Π_1 satisfying bounds of the form (2) was established by Sohler and Woodruff [24]. Here, our contribution is to show that Π_1 can be factored into structured matrices so that the product $\Pi_1 A$ can be computed in $O(nd \log d)$ time. We also remark that, in additional theoretical bounds provided by the FJLT, high-quality numerical implementations of variants of the Hadamard transform exist, which is an additional plus for our empirical evaluations of Theorems 1 and 2.

Remark. Our proof of this theorem uses a tail bound for $\|Bg\|_2^2$ in terms of $\|g\|_2$ and $\|g\|_1$, where g is any positive vector in $\mathbb{R}^n$, and B is the matrix used in our FCT construction. $\|Bg\|_2^2 = \sum_j \gamma_j^2$, where $\gamma_j = \sum_i B_{ji} g_i$ are *anticorrelated* random variables. To get concentration, we independently bounded γ_j^2 in our proof which required $s = r_1^6$ to obtain the high probability result; this resulted in the bound $\kappa = O(d^4 \log^4 d)$.

3.2. FCT2 construction: Via a Fast Johnson–Lindenstrauss transform. This FCT construction first preprocesses by a FJLT and then rescales by Cauchy random variables. Recall that $\delta \in (0, 1]$ is a parameter governing the failure probability of our algorithm; and let $\eta > 0$ be a generic arbitrarily small positive constant (whose value may change from one formula to another). Let $r_1 = c \cdot d \log \frac{d}{\delta}$, $s = c' \cdot (d + \log \frac{n}{\delta})$, and $t = s^{2+\eta}$, where the parameters $c, c' > 0$ are appropriately large constants. Then, we construct $\Pi_1 \in \mathbb{R}^{r_1 \times n}$ as

$$\Pi_1 \equiv \frac{8}{r_1} \sqrt{\frac{\pi t}{2s}} \cdot C\tilde{H},$$

where the following hold:

$C \in \mathbb{R}^{r_1 \times ns/t}$ is a matrix of independent Cauchy random variables.

$\tilde{H} \in \mathbb{R}^{ns/t \times n}$ is a block-diagonal matrix comprising n/t blocks along the diagonal. Each block is the $s \times t$ Fast Johnson–Lindenstrauss matrix G. (The important property that G must satisfy is given by Lemmas 8 and 9 in the proofs below.)

Here, for simplicity, we assume that n/t is an integer.

$$\tilde{H} \equiv \begin{bmatrix} G & & & \\ & G & & \\ & & \ddots & \\ & & & G \end{bmatrix}.$$

Remark. Informally, the matrix $\tilde{H}$ reduces the dimensionality of the input space by a very small amount such that the "slow" Cauchy transform C of [24] can be applied in the allotted time. Then, since we are ultimately multiplying by C, the results of [24] still hold; but since the dimensionality is slightly reduced, the running time is improved.

For this version of the FCT, we have the following theorem. The proof of this theorem may be found in section 3.4.

THEOREM 2 (Fast Cauchy transform (FCT2)). *There is a distribution (given by the above construction) over matrices $\Pi_1 \in \mathbb{R}^{r_1 \times n}$, with $r_1 = O(d \log \frac{d}{\delta})$, such that for arbitrary (but fixed) $A \in \mathbb{R}^{n \times d}$, and for all $x \in \mathbb{R}^d$, the inequalities*

$$\|Ax\|_1 \leq \|\Pi_1 Ax\|_1 \leq \kappa \|Ax\|_1$$

hold with probability $1 - \delta$, where $\kappa = O(\frac{d}{\delta}(d + \log \frac{n}{\delta})^{1+\eta} \log d)$. Further, for any $y \in \mathbb{R}^n$, the product $\Pi_1 y$ can be computed in $O(n \log \frac{d}{\delta})$ time.

Setting δ as a small constant and for $\log n < d$, $r_1 = O(d \log d)$, $\kappa = O(d^{2+\eta} \log d)$, and $\Pi_1 A$ can be computed in $O(nd \log d)$ time. Thus, we have a fast linear oblivious mapping from $\ell_1^n \mapsto \ell_1^{O(d \log d)}$ that has distortion $O(d^{2+\eta} \log d)$ on any (fixed) d-dimensional subspace of $\mathbb{R}^n$.

Remark. For $\log n < d$, FCT2 gives a better dependence of the distortion on d, but more generally FCT2 has a dependence on $\log n$. This dependence arises because the random FJLT matrix does not give a deterministic guarantee for spreading out a vector, whereas the low-coherence matrix used in FCT1 does give a deterministic guarantee. This means that in using the union bound, we need to overcome a factor of n.

Remark. The requirement $t \geq s^{2+\eta}$ is set by the restriction in Lemma 8 in the proof of Theorem 2. In the bound of Theorem 2, $\kappa = \kappa' \sqrt{t}$, where $\kappa' = O(d \log(r_1 d))$ arises from Theorem 3, which originally appeared in [24]. If a stronger version of Lemma 8 can be proved that relaxes the restriction $t \geq s^{2+\eta}$, then correspondingly the bound of Theorem 2 will improve.

Remark. This second construction has the benefit of being easily extended to constructing well-conditioned bases of ℓ_p for $p > 1$; see section 5.

3.3. Proof of Theorem 1 (Fast Cauchy transform (FCT1)).

Preliminaries. Before presenting the proof, we describe the main idea. It follows a similar line of reasoning to [24], and it uses an "uncertainty principle" (which we state as Lemma 6 below).

The uncertainty principle we prove follows from the fact that the concatenation of the Hadamard matrix with the identity matrix is a dictionary of *low coherence*. For background, and arguments similar to those we use in Lemma 6 below, see section 4 of [16]. In particular, see Claim 4.1 and Lemma 4.2 of that section.

To prove the upper bound, we use the existence of a $(d, 1)$-conditioned basis U and apply Π_1 to this basis to show that $\|\Pi_1 Ux\|_1$ cannot expand too much, which in

turn means that $\|\Pi_1 Ax\|_1$ cannot expand too much (for any x). To prove the lower bound, we show that the inequality holds with exponentially high probability for a particular y; and we then use a suitable γ-net to obtain the result for all y.

Main proof. We now proceed with the proof of Theorem 1. We will first prove an upper bound (Proposition 1) and then a lower bound (Proposition 2); the theorem follows by combining Propositions 1 and 2.

PROPOSITION 1. *With probability at least* $1-\delta$, *for all* $x \in \mathbb{R}^d$, $\|\Pi_1 Ax\|_1 \leq \kappa\|Ax\|_1$, *where* $\kappa = O(\frac{d\sqrt{s}}{\delta}\log(r_1 d))$.

Proof. Let $U \in \mathbb{R}^{n\times d}$ be a $(d,1)$-conditioned basis (see Definition 3 below) for the column space of A, which implies that for some $Z \in \mathbb{R}^{d\times d}$ we can write $A = UZ$. Since $\|\Pi_1 Ax\|_1 \leq \kappa\|Ax\|_1$ if and only if $\|\Pi_1 UZx\|_1 \leq \kappa\|UZx\|_1$, it suffices to prove the proposition for U. By construction of U, for any $x \in \mathbb{R}^d$, $\|x\|_\infty \leq \|Ux\|_1$, and so

$$\|\Pi_1 Ux\|_1 \leq \|\Pi_1 U\|_1\|x\|_\infty \leq \|\Pi_1 U\|_1\|Ux\|_1.$$

Thus it is enough to show that $\|\Pi_1 U\|_1 \leq \kappa$. We have

$$\|\Pi_1 U\|_1 = 4\|BC\tilde{H}U\|_1 = 4\sum_{j\in[d]}\|BC\tilde{H}U^{(j)}\|_1 = 4\sum_{j\in[d]}\|BC\hat{U}^{(j)}\|_1,$$

where $\hat{U} \equiv \tilde{H}U$. We will need bounds for $\|\hat{U}^{(j)}\|_1$, for $j \in [d]$, and $\|\hat{U}\|_1$. For any vector $y \in \mathbb{R}^n$, we represent y by its n/s blocks of size s, so $z_i \in \mathbb{R}^s$ and $y^T = [z_1^T, z_2^T, \ldots, z_{n/s}^T]$. Recall that $G_s \equiv \left[\begin{smallmatrix} H_s \\ I_s \end{smallmatrix}\right]$, and observe that $\|G_s\|_2 = \sqrt{2}$. By explicit calculation,

$$\|\tilde{H}y\|_1 = \sum_{i\in[n/s]}\|G_s z_i\|_1.$$

Since $\|G_s z_i\|_1 \leq \sqrt{2s}\|G_s z_i\|_2 \leq \sqrt{4s}\|z_i\|_2 \leq \sqrt{4s}\|z_i\|_1$, it follows that

$$\|\tilde{H}y\|_1 \leq \sqrt{4s}\sum_{i\in[n/s]}\|z_i\|_1 = \sqrt{4s}\|y\|_1.$$

Applying this to $y = U^{(j)}$ for $j \in [d]$ yields

$$\|\hat{U}^{(j)}\|_1 \leq \sqrt{4s}\|U^{(j)}\|_1 \qquad \text{and} \qquad \|\hat{U}\|_1 = \sum_{j\in[d]}\|\hat{U}^{(j)}\|_1 \leq \sqrt{4s}\|U\|_1 \leq d\sqrt{4s}, \tag{3}$$

since $\|U\|_1 \leq d$ because U is $(d,1)$-conditioned.

The (i,j) entry of $BC\hat{U}$ is $\sum_k B_{ik}C_{kk}\hat{U}_{kj}$, which is a Cauchy scaled by $\gamma_{ij} = \sum_k |B_{ik}\hat{U}_{kj}|$. So,

$$\|BC\hat{U}\|_1 = \sum_{i\in[r_1],j\in[d]}\gamma_{ij}\tilde{C}_{ij},$$

where $\tilde{C}_{ij}$ are *dependent* Cauchy random variables. Using $\sum_i B_{ik} = 1$, we obtain

$$\sum_{i,j}\gamma_{ij} = \sum_{i,j,k}|B_{ik}\hat{U}_{kj}| = \sum_{j,k}|\hat{U}_{kj}|\sum_i B_{ik} = \sum_{j,k}|\hat{U}_{kj}| = \|\hat{U}\|_1.$$

Hence, we can apply Lemma 3 with $\gamma = \|\hat{U}\|_1$ and $m = r_1 d$ to obtain

$$\mathbf{Pr}\left[\|BC\hat{U}\|_1 > t\|\hat{U}\|_1\right] \leq \frac{(\log(r_1 d) + \log t)}{t}\left(1 + o(1)\right).$$

Setting the right-hand side (RHS) to δ, it suffices that $t = O(\frac{1}{\delta}\log(dr_1))$. Thus, with probability at least $1-\delta$,

$$\|BC\hat{U}\|_1 = O\left(\frac{1}{\delta}\log(dr_1)\|\hat{U}\|_1\right) = O\left(\frac{d}{\delta}\sqrt{s}\log(dr_1)\right). \qquad \square$$

Before we prove the lower bound, we need the following lemma, which is derived using a sparsity result for matrices with unit norm rows and low "coherence," as measured by the maximum magnitude of the inner product between distinct rows (G_s is a matrix with low coherence). This result mimics results in [9, 8, 13, 27, 16].

LEMMA 6. *For* $G = \left[\begin{smallmatrix} H_s \\ I_s \end{smallmatrix}\right]$ *and any* $z \in \mathbb{R}^s$, $\|Gz\|_1 \geq \frac{1}{2}s^{1/4}\|z\|_2$.

Proof. We can assume $\|z\|_2 = 1$, and so $\|Gz\|_2^2 = 2$. Let $G_{(S')}$ be k rows of G, with κ of them coming from H_s and $k-\kappa$ from I_s. $G_{(S')}G_{(S')}^T = I + \Lambda$, where Λ is a symmetric 2×2 block matrix

$$\begin{bmatrix} \mathbf{0} & \frac{1}{\sqrt{s}}Q \\ \frac{1}{\sqrt{s}}Q^T & \mathbf{0} \end{bmatrix},$$

where the entries in $Q \in \mathbb{R}^{\kappa\times(k-\kappa)}$ are ± 1, and so $\|Q\|_2 \leq \sqrt{\kappa(k-\kappa)} \leq \frac{1}{2}k$.

$$\|G_{(S')}\|_2^2 = \|G_{(S')}G_{(S')}^T\|_2 \leq 1 + \|\Lambda\|_2 = 1 + \frac{1}{\sqrt{s}}\|Q\|_2 \leq 1 + \frac{k}{2\sqrt{s}}.$$

Now, given any z, we set $k = 2\beta\sqrt{s}$ with $\beta = \frac{2}{5}$, and choose $G_{(S')}$ to be the rows corresponding to the k components of Gz having largest magnitude, with $G_{(S)}$ being the rows with indices in $[s]\setminus S'$. Then, $\|G_{(S')}z\|_2^2 \leq 1+\beta$, and so the entry in $G_{(S')}z$ with smallest magnitude has magnitude at most $a = \sqrt{(1+\beta)/k} = \sqrt{(1+\beta)/2\beta}s^{-1/4}$. We now consider $G_{(S)}z$. Since $\|Gz\|_2^2 = 2$, $\|G_{(S)}z\|_2^2 \geq 1-\beta$; further, all components have magnitude at most a (as all the components of $G_{(S)}z$ have smaller magnitude than those of $G_{(S')}z$). $\|G_{(S)}z\|_1$ is minimized by concentrating all the entries into as few components as possible. Since the number of nonzero components is at least $(1-\beta)/a^2 = 2\beta s^{1/2}(1-\beta)/(1+\beta)$, giving these entries the maximum possible magnitude results in

$$\|G_{(S)}z\|_1 \geq a \times \frac{(1-\beta)}{a^2} = (1-\beta)\sqrt{2\beta(1+\beta)}s^{1/4} \geq 0.63s^{1/4}$$

(where we used $\beta = \frac{2}{5}$). We are done because $\|Gz\|_1 \geq \|G_{(S)}z\|_1$. $\square$

We now prove the lower bound. We assume that Proposition 1 holds for Π_1, which is true with probability at least $1-\delta$ for κ as defined in Proposition 1. Then, by a union bound, Propositions 1 and 2 both hold with probability at least

$$1 - \delta - \exp\left(-\frac{r_1}{48} + d\log(2d\kappa)\right) - \exp\left(-\frac{s^{1/2}}{8r_1^2} + \log r_1 + d\log(2d\kappa)\right)$$

(δ and κ are from Proposition 1). Since $s^{1/2} = r_1^3$, by choosing $r_1 = \alpha d\log\frac{d}{\delta}$ for large enough α, the final probability of failure is at most 2δ because $\kappa = O(\frac{d\sqrt{s}}{\delta}\log(r_1 d)) = O(\text{poly}(d))$.

PROPOSITION 2. *Assume Proposition* 1 *holds. Then, for all* $x \in \mathbb{R}^d$, $\|\Pi_1 Ax\|_1 \geq \|Ax\|_1$ *holds with probability at least*

$$1 - \exp\left(-\frac{r_1}{48} + d\log(2d\kappa)\right) - \exp\left(-\frac{s^{1/2}}{8r_1^2} + \log r_1 + d\log(2d\kappa)\right).$$

Proof. First we will show a result for fixed $y \in \mathbb{R}^n$, summarized in the next lemma.

LEMMA 7. $\mathbf{Pr}\left[\|\Pi_1 y\|_1 < 2\|y\|_1\right] \leq \exp\left(-\frac{r_1}{48}\right) + \exp\left(-\frac{s^{1/2}}{8r_1^2} + \log r_1\right)$.

Given this lemma, the proposition follows by putting a γ-net Γ on the range of A (observe that the range of A has dimension at most d). This argument follows the same line as in sections 3 and 4 of [24]. Specifically, let L be any fixed (at most) d-dimensional subspace of $\mathbb{R}^n$ (in our case, L is the range of A). Consider the γ-net on L with cubes of side γ/d. There are $(2d/\gamma)^d$ such cubes required to cover the hypercube $\|y\|_\infty \leq 1$; and, for any two points y_1, y_2 inside the same γ/d-cube, $\|y_1 - y_2\|_1 \leq \gamma$. From each of the γ/d-cubes, select a fixed representative point which we will generically refer to as y^*; select the representative to have $\|y^*\|_1 = 1$ if possible. By a union bound and Lemma 7,

$$\mathbf{Pr}\left[\min_{y^*} \|\Pi_1 y^*\|_1 / \|y^*\|_1 < 2\right] \leq (2d/\gamma)^d \left(\exp\left(-\frac{r_1}{48}\right) + \exp\left(-\frac{s^{1/2}}{8r_1^2} + \log r_1\right)\right).$$

We will thus condition on the high probability event that $\|\Pi_1 y^*\|_1 \geq 2\|y^*\|$ for all y^*. For any $y \in L$ with $\|y\|_1 = 1$, let y^* denote the representative point for the cube in which y resides ($\|y^*\|_1 = 1$ as well). Then, $\|y - y^*\| \leq \gamma$.

$$\|\Pi_1 y\|_1 = \|\Pi_1 y^* + \Pi_1(y - y^*)\|_1 \geq \|\Pi_1 y^*\|_1 - \|\Pi_1(y - y^*)\|_1 \geq 2\|y^*\|_1 - \kappa\|y - y^*\|_1,$$

where the last inequality holds using Proposition 1. By choosing $\gamma = 1/\kappa$ and recalling that $\|y^*\|_1 = 1$, we have that $\|\Pi_1 y\|_1 \geq 1$, with probability at least

$$1 - \exp(d\log(2d\kappa)) \left(\exp\left(-\frac{r_1}{48}\right) + \exp\left(-\frac{s^{1/2}}{8r_1^2} + \log r_1\right)\right).$$

All that remains is to prove Lemma 7. As in the proof of Proposition 1, we represent any vector $y \in \mathbb{R}^n$ by its n/s blocks of size s, so $z_i \in \mathbb{R}^s$ and $y^T = [z_1^T, z_2^T, \ldots, z_{n/s}^T]$. Let $g = \tilde{H}y$, and

$$g = \begin{bmatrix} G_s z_1 \\ G_s z_2 \\ \vdots \\ G_s z_{n/s} \end{bmatrix}.$$

We have that $\|g\|_2^2 = \sum_i \|G_s z_i\|_2^2 = 2\sum_i \|z_i\|_2^2 = 2\|y\|_2^2$, and

$$\|g\|_1 = \sum_i \|G_s z_i\|_1 \geq \frac{1}{2} s^{1/4} \sum_i \|z_i\|_2 \geq \frac{1}{2} s^{1/4} \left(\sum_i \|z_i\|_2^2\right)^{1/2} = \frac{1}{2} s^{1/4} \|y\|_2.$$

We conclude that $\|g\|_1 \geq \frac{1}{2\sqrt{2}} s^{1/4} \|g\|_2$, which intuitively means that g is "spread out." We now analyze $\|BCg\|_1$. (Recall that $\Pi_1 y = 4BCg$, where $g = \tilde{H}y$.)

$$(BCg)_j = \sum_{i=1}^{2n} B_{ji} C_{ii} g_i$$

is a Cauchy random variable $\tilde{C}_j$ scaled by $\gamma_j = \sum_{i=1}^{2n} B_{ji}|g_i|$. Further, because each column of B has exactly one nonzero element, the $\tilde{C}_j$ for $j \in [r_1]$ are independent.

Thus, the random variables $\|BCg\|_1$ and $\sum_{j\in[r_1]} |\tilde{C}_j|\gamma_j$ have the same distribution. To apply Lemma 4, we need to bound $\sum_j \gamma_j$ and $\sum_j \gamma_j^2$. First,

$$\sum_{j\in[r_1]} \gamma_j = \sum_{j\in[r_1]} \sum_{i\in[n]} B_{ji}|g_i| = \sum_{i\in[n]} |g_i| \sum_{j\in[r_1]} B_{ji} = \sum_{i=1}^{2n} |g_i| = \|g\|_1,$$

where the last inequality is because $B^{(i)}$ is a standard basis vector. To bound $\sum_j \gamma_j^2$, we will show that γ_j is nearly uniform. Since γ_j is a weighted sum of independent Bernoulli random variables (because B_{ji} and B_{jk} are independent for $i \neq k$), we can use Lemma 2 with $\xi_i = |g|_i$ and $1 - p = 1 - 1/r_1 \leq 1$, and so $\sum_i \xi_i = \|g\|_1$ and $\sum_i \xi_i^2 = \|g\|_2^2$; setting $t = 1/r_1$ in Lemma 2,

$$\mathbf{Pr}\left[\gamma_j \geq \frac{2\|g\|_1}{r_1}\right] \leq \exp\left(-\frac{\|g\|_1^2}{2\|g\|_2^2 r_1^2}\right) \leq \exp\left(-\frac{s^{1/2}}{8r_1^2}\right).$$

By a union bound, none of the γ_j exceeds $2\|g\|_1/r_1$ with probability at most $r_1 \exp\left(-s^{1/2}/8r_1^2\right)$. We assume this high probability event, in which case $\sum_j \gamma_j^2 \leq 4\|g_1\|_1^2/r_1$. We can now apply Lemma 4 with $\beta^2 = r_1/4$ and $t = \frac{1}{2}$ to obtain

$$\mathbf{Pr}\left[\sum_{j\in[r_1]} |\tilde{C}_j|\gamma_j \leq \tfrac{1}{2}\|g\|_1\right] \leq \exp\left(-\frac{r_1}{48}\right).$$

By a union bound, $\|BCg\|_1 \geq \frac{1}{2}\|g\|_1$ with probability at least $1 - \exp\left(-\frac{r_1}{48}\right) - \exp\left(-\frac{s^{1/2}}{8r_1^2} + \log r_1\right)$. Scaling both sides by 4 gives the lemma. ∎

Running time. The running time follows from the time to compute the product $H_s x$ for a Hadamard matrix H_s, which is $O(s \log s)$ time. The time to compute $\tilde{H}y$ is dominated by n/s computations of $H_s z_i$, which is a total of $O(\frac{n}{s} \cdot s \log s) = O(n \log s)$ time. Since C is diagonal, premultiplying by C is $O(n)$ and further premultiplying by B takes time $O(nnz(B))$, the number of nonzero elements in B (which is $2n$). Thus the total time is $O(n \log s + n) = O(n \log r_1)$ as desired.

3.4. Proof of Theorem 2 (Fast Cauchy transform (FCT2)).

Preliminaries. We will need results from prior work, which we paraphrase in our notation.

DEFINITION 2 (Definition 2.1 of [2]). *For $\varepsilon \in (0, \frac{1}{2}]$, a distribution on $s \times t$ real matrices G ($s \leq t$) has the* Manhattan Johnson–Lindenstrauss property *(MJLP) if for any (fixed) vector $x \in \mathbb{R}^t$ the inequalities*

$$(1-\varepsilon)\|x\|_2 \leq \|Gx\|_2 \leq (1+\varepsilon)\|x\|_2,$$
$$c_3\sqrt{s}(1-\varepsilon)\|x\|_2 \leq \|Gx\|_1 \leq c_3\sqrt{s}(1+\varepsilon)\|x\|_2$$

hold with probability at least $1 - c_1 e^{-c_2 k\varepsilon^2}$ (with respect to G) for global constants $c_1, c_2, c_3 > 0$.

Remark. This is the standard Johnson–Lindenstrauss property with the additional requirement on $\|Gx\|_1$. Essentially it says that Gx is a nearly uniform, so that $\|Gx\|_1 \approx \sqrt{s}\|Gx\|_2 \approx \sqrt{s}\|x\|_2$.

LEMMA 8 (Theorem 2.2 of [2]). *Let $\eta > 0$ be an arbitrarily small constant. For any s, t satisfying $s \le t^{1/2-\eta}$, there exists an algorithm that constructs a random $s \times t$ matrix G that is sampled from an MJLP distribution with $c_3 = \sqrt{\frac{2}{\pi}}$. Further, the time to compute Gx for any $x \in \mathbb{R}^t$ is $O(t \log s)$.*

We will need these lemmas to get a result for how an arbitrary subspace L behaves under the action of G, extending Lemma 8 to every $x \in L$, not just a fixed x. In the next lemma, the 2-norm bound can be derived using Lemma 8 (above) and Theorem 19 of [18] by placing a γ-net on L and bounding the size of this γ-net. (See Lemma 4 of [3].) The Manhattan norm bound is then derived using a second γ-net argument together with an application of the 2-norm bound. The constants c_1, c_2, and c_3 in this lemma are from Definition 2; and the G in Lemmas 8 and 9, with the constants c_1, c_2, and c_3 from Definition 2, is the same G used in our FCT2 construction for $\tilde{H}$. We present the complete proof of Lemma 9 in Appendix H.

LEMMA 9. *Let L be any (fixed) d-dimensional subspace of $\mathbb{R}^t$, and let G be an $s \times t$ matrix sampled from a distribution having the MJLP property. Given $\varepsilon \in (0, \frac{1}{3}]$, let $s = 36(k + \frac{8d}{c_3\varepsilon} + \log(2c_1))/c_2\varepsilon^2 = O(\frac{k}{\varepsilon^2} + \frac{d}{\varepsilon^3})$. Then, with probability at least $1 - e^{-k}$, for every $x \in \mathbb{R}^t$,*

$$\sqrt{1-\varepsilon}\|x\|_2 \le \|Gx\|_2 \le \sqrt{1+\varepsilon}\|x\|_2,$$
$$c_3\sqrt{s}(1-\varepsilon)\|x\|_2 \le \|Gx\|_1 \le c_3\sqrt{s}(1+\varepsilon)\|x\|_2.$$

We also need a result on how the matrix of Cauchy random variables C behaves when it hits a vector y. The next theorem is Theorem 5 of [24]. For completeness and also to fix some minor errors in the proof of [24], we give a proof of Theorem 3 in Appendix G.

THEOREM 3. *Let L be an arbitrary (fixed) subspace of $\mathbb{R}^n$ having dimension at most d, and let C be an $r_1 \times n$ matrix of i.i.d. Cauchy random variables with $r_1 = c \cdot d \log \frac{d}{\delta}$ for large enough constant c. Then, with probability at least $1 - \delta$, and for all $y \in L$,*

$$\|y\|_1 \le \tfrac{4}{r_1}\|Cy\|_1 \le \kappa'\|y\|_1,$$

where $\kappa' = O(\frac{d}{\delta}\log(r_1 d))$.

Note that for δ fixed to some small error probability, $r_1 = O(d \log d)$, and the product Cy in the theorem above can be computed in time $O(r_1 n) = O(nd \log d)$.

Main proof. We now proceed with the proof of Theorem 2. We need to analyze the product $C\tilde{H}Ax$ for all $x \in \mathbb{R}^d$. Let $y = Ax \in \mathbb{R}^n$, so that $y \in \operatorname{colsp} A \equiv \{Az \mid z \in \mathbb{R}^d\}$, and the column space $\operatorname{colsp} A$ is a d-dimensional subspace of $\mathbb{R}^n$. Partition the coordinate set $[n]$ into n/t contiguous groups of t coordinates. We will work with the block representation of y, as defined by this partition, i.e., with $y^T = [z_1^T, z_2^T, \ldots, z_{n/t}^T]$, where $z_i = A_{(\{i\})}x$ and where $A_{(\{i\})}$ is the block of t rows in A corresponding to the indices in z_i. Then,

$$\tilde{H}y = \begin{bmatrix} Gz_1 \\ Gz_2 \\ \vdots \\ Gz_{n/t} \end{bmatrix}.$$

The vector $z_i \in \operatorname{colsp} A_{(\{i\})}$, noting that $\operatorname{colsp} A_{(\{i\})}$ is a subspace of $\mathbb{R}^t$ of dimension at most d. Let $U_i \in \mathbb{R}^{t \times d}$ be an orthonormal basis for $\operatorname{colsp} A_{(\{i\})}$, and let $z_i = U_i w_i$. Setting $\varepsilon = \frac{1}{2}$ in Lemma 9, and recalling that G is $s \times t$, k in Lemma 9 can be expressed

as $k = \frac{c_2 s}{144} - \frac{16d}{c_3} - \log(2c_2)$. Applying a union bound, we have for all $i \in [n/t]$ with probability at least $1 - 2c_1 \cdot \frac{n}{t} \cdot \exp(-\frac{c_2 s}{144} + \frac{16d}{c_3})$ that for all $y = Ax$ (and corresponding $z_i \in \mathbb{R}^t$), it holds that

$$\sqrt{\tfrac{1}{2}}\|z_i\|_2 \le \|Gz_i\|_2 \le \sqrt{\tfrac{3}{2}}\|z_i\|_2,$$
$$\tfrac{1}{2}c_3\sqrt{s}\|z_i\|_2 \le \|Gz_i\|_1 \le \tfrac{3}{2}c_3\sqrt{s}\|z_i\|_2.$$

We will condition on this event, which occurs with high probability for the given parameters. We can now bound $\|\tilde{H}y\|_1 = \sum_{i\in[n/t]} \|Gz_i\|_1$ as follows:

(4)
$$\|\tilde{H}y\|_1 = \sum_{i\in[n/t]} \|Gz_i\|_1 \le \tfrac{3}{2}c_3\sqrt{s} \sum_{i\in[n/t]} \|z_i\|_2 \le \tfrac{3}{2}c_3\sqrt{s} \sum_{i\in[n/t]} \|z_i\|_1 = \tfrac{3}{2}c_3\sqrt{s}\|y\|_1;$$

(5)
$$\|\tilde{H}y\|_1 = \sum_{i\in[n/t]} \|Gz_i\|_1 \ge \tfrac{1}{2}c_3\sqrt{s} \sum_{i\in[n/t]} \|z_i\|_2 \ge \tfrac{1}{2}c_3\sqrt{\tfrac{s}{t}} \sum_{i\in[n/t]} \|z_i\|_1 = \tfrac{1}{2}c_3\sqrt{\tfrac{s}{t}}\|y\|_1.$$

Since $\operatorname{colsp}\tilde{H}A$ has dimension at most d, we can apply Theorem 3 to it. We have that with probability at least $1 - \delta$, for all $x \in \mathbb{R}^d$,

$$\|\tilde{H}Ax\|_1 \le \frac{4}{r_1}\|C\tilde{H}Ax\|_1 \le \kappa'\|\tilde{H}Ax\|_1,$$

where $\kappa' = O(\frac{d}{\delta}\log(r_1 d))$ from Theorem 3. Recall that $\Pi_1 \equiv \frac{8}{r_1}\sqrt{\frac{\pi t}{2s}}C\tilde{H}$. We now use (4) and (5) with $y = Ax$, and after multiplying by $\frac{2}{c_3}\sqrt{\frac{t}{s}}$ and setting $c_3 = \sqrt{2/\pi}$, we conclude that for all $x \in \mathbb{R}^d$,

$$\|Ax\|_1 \le \|\Pi_1 Ax\|_1 \le 3\kappa'\sqrt{t}\|Ax\|_1 \tag{6}$$

holds with probability at least $1 - \delta - 2c_1 \cdot \frac{n}{t}\exp(-\frac{c_2 s}{144} + \frac{16d}{c_3}) \ge 1 - 2\delta$ (by choosing $s \ge \frac{144}{c_2}(\frac{16d}{c_3} + \log\frac{2c_1 n}{\delta t})$). The theorem follows because $\log n \le d$, and hence $\kappa' = O(\frac{d}{\delta}\log d)$, $s = O(d + \log\frac{1}{\delta})$, and $t = O(s^{2+\eta})$.

Running time. We now evaluate the time to compute $\Pi_1 y$ for $y \in \mathbb{R}^n$. We first compute $\tilde{H}y$, which requires n/t computations of Gz_i. Since $s = t^{1/2-\eta/2}$, we can invoke Lemma 8. The time to compute all Gz_i is $\frac{n}{t} \cdot t\log s = n\log s$. Since $\tilde{H}y$ is $(ns/t)\times 1$, it takes $O(r_1 ns/t)$ time to compute $C\tilde{H}y$, which concludes the computation. The total running time is $O(n\log s + nr_1 s/t)$. Using $\log n \le d$, $s = O(d)$, $t = s^{2+\eta}$, $r_1 = O(d\log\frac{d}{\delta})$ we need total time $O(n\log\frac{d}{\delta})$. To compute $\Pi_1 A$, we need to compute $\Pi_1 A^{(j)}$ for d vectors $A^{(j)}$, resulting in a total run time $O(nd\log\frac{d}{\delta})$.

4. Algorithmic applications in ℓ_1 of the FCT. In this section, we describe three related applications of the FCT to ℓ_1-based problems. The first is to the fast construction of an ℓ_1 well-conditioned basis and the fast approximation of ℓ_1 leverage scores; the second is a fast algorithm for the least absolute deviations or ℓ_1 regression problem; and the third is to a fast algorithm for the ℓ_1 norm subspace approximation problem.

FastL1Basis(A):
1: Let Π_1 be an $r_1 \times n$ matrix satisfying (7), e.g., as constructed with one of the FCTs of section 3.
2: Compute $\Pi_1 A \in \mathbb{R}^{r_1 \times d}$ and its QR factorization: $\Pi_1 A = QR$, where Q is an orthogonal matrix, i.e., $Q^T Q = I$.
3: Return $U = AR^{-1} = A(Q^T \Pi_1 A)^{-1}$.

FIG. 1. *Our main algorithm for the fast construction of an ℓ_1 well-conditioned basis of an $n \times d$ matrix A. Note the structural similarities with the algorithm of* [10] *for computing quickly approximations to the ℓ_2 leverage scores and an ℓ_2 well-conditioned basis.*

4.1. Fast construction of an ℓ_1 well-conditioned basis and ℓ_1 leverage scores. We start with the following definition, adapted from [7], of a basis that is "good" for the ℓ_1 norm in a manner that is analogous to how an orthogonal matrix is "good" for the ℓ_2 norm.

DEFINITION 3 (ℓ_1 well-conditioned basis (adapted from [7])). *A basis U for the range of A is (α, β)-conditioned if $\|U\|_1 \leq \alpha$ and for all $x \in \mathbb{R}^d$, $\|x\|_\infty \leq \beta \|Ux\|_1$. We will say that U is* well-conditioned *if α and β are low-degree polynomials in d, independent of n.*

Remark. An Auerbach basis for A is $(d, 1)$-conditioned, and thus we know that there exist well-conditioned bases for ℓ_1. More generally, well-conditioned bases can be defined in any ℓ_p norm, using the notion of a dual norm ℓ_p^*, and these have proven important for solving ℓ_p regression problems [7]. Our focus in this section is the ℓ_1 norm, for which the dual norm is the ℓ_∞ norm, but in section 5 we will return to a discussion of extensions to the ℓ_p norm.

Our main algorithm for constructing an ℓ_1 well-conditioned basis, FastL1Basis, is summarized in Figure 1. This algorithm was originally presented in [24], and our main contribution here is to improve its running time. We note that in step 3, we do not explicitly compute the product of A and R^{-1}, but rather just return A and R^{-1} with the promise that AR^{-1} is well-conditioned. The leading-order term in our running time to compute R^{-1} is $O(nd \log d)$, while in [24] it is $O(nd^2)$, or with fast matrix multiplication, $O(nd^{1.376})$.

Given an $n \times d$ matrix A, let $\Pi_1 \in \mathbb{R}^{r_1 \times n}$ be any projection matrix such that for any $x \in \mathbb{R}^d$,

$$\|Ax\|_1 \leq \|\Pi_1 Ax\|_1 \leq \kappa \|Ax\|_1. \tag{7}$$

For example, it could be constructed with either of the FCT constructions described in section 3, or with the "slow" Cauchy transform of [24], or via some other means. After computing the matrix Π_1, the FastL1Basis algorithm of Figure 1 consists of the following steps: construct $\Pi_1 A$ and an R such that $\Pi_1 A = QR$, where Q has orthonormal columns (for example, using a QR factorization of $\Pi_1 A$); and then return $U = AR^{-1} = A(Q^T \Pi_1 A)^{-1}$.

The next theorem and its corollary are our main results for the FastL1Basis algorithm; and this theorem follows by combining our Theorem 2 with Theorems 9 and 10 of [24]. The proof of this theorem may be found in Appendix B.

THEOREM 4 (fast ℓ_1 well-conditioned basis). *For any $A \in \mathbb{R}^{n \times d}$, the basis $U = AR^{-1}$ constructed by FastL1Basis(A) of Figure 1 using any Π_1 satisfying (7) is a $(d\sqrt{r_1}, \kappa)$-conditioned basis for the range of A.*

COROLLARY 1. *If Π_1 is obtained from the FCT2 construction of Theorem 2, then the resulting U is an (α, β)-conditioned basis for A, with $\alpha = O(d^{3/2}\log^{1/2} d)$ and $\beta = O(d^{2+\eta}\log d)$, with probability $1-\delta$. The time to compute the change of basis matrix R^{-1} is $O(nd\log d + d^3 \log d)$, assuming $\log n = O(d)$ and $\delta > 0$ is a fixed constant.*

Remark. Our constructions that result in Π_1 satisfying (7) do not require that $A \in \mathbb{R}^{n\times d}$; they only require that A have rank d and hence can be applied to any $A \in \mathbb{R}^{n\times m}$ having rank d. In this case, a small modification is needed in the construction of U, because $R \in \mathbb{R}^{d\times m}$, and so we need to use $R^\dagger$ instead of R^{-1}. The running time will involve terms with m. This can be improved by processing A quickly into a smaller matrix by sampling columns so that the range is preserved (as in [24]), which we do not discuss further.

The notion of a well-conditioned basis plays an important role in our subsequent algorithms. Basically, the reason is that these algorithms compute approximate answers to the problems of interest (either the ℓ_1 regression problem or the ℓ_1 subspace approximation problem) by using information in that basis to construct a nonuniform importance sampling distribution with which to randomly sample. This motivates the following definition.

DEFINITION 4 (ℓ_1 leverage scores). *Given a well-conditioned basis U for the range of A, let the n-dimensional vector $\tilde{\lambda}$, with elements defined as $\tilde{\lambda}_i = ||U_{(i)}||_1$, be the* ℓ_1 leverage scores *of A.*

Remark. The name ℓ_1 *leverage score* is by analogy with the ℓ_2 *leverage scores*, which are important in random sampling algorithms for ℓ_2 regression and low-rank matrix approximation [21, 20, 10]. As with ℓ_2 regression and low-rank matrix approximation, our result for ℓ_1 regression and ℓ_1 subspace approximation will ultimately follow from the ability to approximate these scores quickly. Note, though, that these ℓ_1-based scores are not well-defined for a given matrix A, in the sense that the ℓ_1 norm is not rotationally invariant, and thus depending on the basis that is chosen, these scores can differ by factors that depend on low-degree polynomials in d. This contrasts with ℓ_2, since for ℓ_2 any orthogonal matrix spanning a given subspace leads to the same ℓ_2 leverage scores. We will tolerate this ambiguity since these ℓ_1 leverage scores will be used to construct an importance sampling distribution, and thus up to low-degree polynomial factors in d, which our analysis will take into account, it will not matter.

4.2. Fast ℓ_1 regression. Here, we consider the ℓ_1 regression problem, also known as the *least absolute deviations* problem, the goal of which is to minimize the ℓ_1 norm of the residual vector $Ax - b$. That is, given as input a design matrix $A \in \mathbb{R}^{n\times d}$, with $n > d$, and a response or target vector $b \in \mathbb{R}^n$, compute

$$\min_{x\in\mathbb{R}^d} ||Ax - b||_1 \tag{8}$$

and an x^* achieving this minimum. We start with our main algorithm and theorem for this problem; and we then describe how a somewhat more sophisticated version of the algorithm yields improved running time bounds.

4.2.1. Main algorithm for fast ℓ_1 regression. Prior work has shown that there is a diagonal sampling matrix D with a small number of nonzero entries so that $\hat{x} = \text{argmin}_{x\in\mathbb{R}^d}\|D(Ax-b)\|_1$ satisfies

$$\|A\hat{x} - b\|_1 \le (1+\varepsilon)\|Ax^* - b\|_1,$$

FastCauchyRegression(A, b):

1: Let $X = \begin{bmatrix} A & -b \end{bmatrix} \in \mathbb{R}^{n \times (d+k)}$, and construct Π_1, an $r_1 \times n$ matrix satisfying (7) with A replaced by X. (If b is a vector, then $k = 1$.)
2: Compute $X' = \Pi_1 X \in \mathbb{R}^{r_1 \times (d+k)}$ and its QR factorization, $\Pi_1 X = QR$. (Note that $\Pi_1 X R^{-1}$ has orthonormal columns.)
3: Let $\Pi_2 \in \mathbb{R}^{(d+k) \times r_2}$ be a matrix of independent Cauchys, with $r_2 = 15 \log \frac{2n}{\delta}$.
4: Let $U = XR^{-1}$, and construct $\Lambda = U\Pi_2 \in \mathbb{R}^{n \times r_2}$.
5: For $i \in [n]$, compute $\lambda_i = \text{median}_{j \in [r_2]} |\Lambda_{ij}|$.
6: For $i \in [n]$ and $s = \frac{63\kappa(d+k)\sqrt{r_1}}{\varepsilon^2}\big((d+k)\log\frac{24\kappa(d+k)\sqrt{r_1}}{\varepsilon} + \log\frac{2}{\delta}\big)$, compute probabilities
$$\hat{p}_i = \min\left\{1, s \cdot \frac{\lambda_i}{\sum_{i \in [n]} \lambda_i}\right\}.$$
7: Let $D \in \mathbb{R}^{n \times n}$ be diagonal with independent entries: $D_{ii} = \begin{cases} \frac{1}{\hat{p}_i}, & \text{prob. } \hat{p}_i; \\ 0, & \text{prob. } 1 - \hat{p}_i. \end{cases}$
8: Return $\hat{x} \in \mathbb{R}^d$ that minimizes $\|DAx - Db\|_1$ with respect to x (using linear programming).

FIG. 2. *Algorithm for solving ℓ_1 regression. Note that in step 6, we sample rows of A and b so that the expected number of rows sampled is at most s. Instead of this independent sampling (without replacement), we could sample exactly s rows independently* with *replacement according to the probabilities $\hat{p}_i = \lambda_i / \sum_{i \in [n]} \lambda_i$, and all our results continue to hold up to small factors.*

where x^* is an optimal solution for the minimization in (8); see [7, 24]. The matrix D can be found by sampling its diagonal entries independently according to a set of probabilities p_i that are proportional to the ℓ_1 leverage scores of A. Here, we give a fast algorithm to compute estimates $\hat{p}_i$ of these probabilities. This permits us to develop an improved algorithm for ℓ_1 regression and to construct efficiently a small coreset for an arbitrary ℓ_1 regression problem.

In more detail, Figure 2 presents the FastCauchyRegression algorithm, which we summarize here. Let $X = \begin{bmatrix} A & -b \end{bmatrix}$. First, a matrix Π_1 satisfying (7) is used to reduce the dimensionality of X to $\Pi_1 X$ and to obtain the orthogonalizer R^{-1}. Let $U = XR^{-1}$ be the resulting well-conditioned basis for the range of X. The probabilities we use to sample rows are essentially the row norms of U. However, to compute XR^{-1} explicitly takes $O(nd^2)$ time, which is already too costly, and so we need to estimate $\|U_{(i)}\|_1$ without explicitly computing U. To construct these probabilities quickly, we use a second random projection Π_2—on the *right*. This second projection allows us to estimate the norms of the rows of XR^{-1} efficiently to within relative error (which is all we need) using the median of r_2 independent Cauchys, each scaled by $||U_{(i)}||_1$. (Note that this is similar to what was done in [10] to approximate the ℓ_2 leverage scores of an input matrix.) These probabilities are then used to construct a carefully downsampled (and rescaled) problem, the solution to which will give us our $(1 + \varepsilon)$ approximation to the original problem.

The next theorem summarizes our main quality-of-approximation results for the FastCauchyRegression algorithm of Figure 2. It improves the $O(nd^2 + \text{poly}(d\varepsilon^{-1} \log n))$ algorithm of [24], which in turn improved the result in [7]. (Technically, the running time of [24] is $O(nd^{\omega^+ - 1} + \text{poly}(d\varepsilon^{-1} \log n))$, where ω^+ is any constant larger than the

exponent for matrix multiplication; for practical purposes, we can set $\omega^+ = 3$.) Our improved running time comes from using the FCT and a simple row norm estimator for the row norms of a well-conditioned basis. The proof of this theorem may be found in Appendix C.

THEOREM 5 (Fast Cauchy ℓ_1 regression). *Given are $\varepsilon \in (0,1)$, $\rho > 0$, $A \in \mathbb{R}^{n\times d}$, and $b \in \mathbb{R}^n$. FastCauchyRegression(A,b) constructs a coreset specified by the diagonal sampling matrix D and a solution vector $\hat{x} \in \mathbb{R}^d$ that minimizes the weighted regression objective $\|D(Ax-b)\|_1$. The solution $\hat{x}$ satisfies, with probability at least $1-\frac{1}{d^\rho}$ ($\rho > 0$ is a constant),*

$$\|A\hat{x} - b\|_1 \le \left(\frac{1+\varepsilon}{1-\varepsilon}\right)\|Ax-b\|_1 \qquad \forall x \in \mathbb{R}^d.$$

Further, with probability $1-o(1)$, the entire algorithm to construct $\hat{x}$ runs in time

$$O\left(nd\log n + \phi(s,d)\right) = O\left(nd\log n + \tfrac{1}{\varepsilon^2}\operatorname{poly}(d, \log\tfrac{d}{\varepsilon})\right),$$

where $\phi(s,d)$ is the time to solve an ℓ_1 regression problem on s vectors in d dimensions, and if FCT2 is used to construct Π_1 then $s = O\big(\frac{1}{\varepsilon^2} d^{\rho+\frac{9}{2}+\eta}\log^{\frac{9}{2}}(\frac{d}{\varepsilon})\big)$.

Remarks. Several remarks about our results for the ℓ_1 regression problem are in order:

- Our proof analyzes a more general problem $\min_{x\in\mathcal{C}} ||Xx||_1$, where $\mathcal{C} \subseteq \mathbb{R}^d$ is a convex set. In order to get the result, we need to preserve norms under sampling, which is what Lemma 5 allows us to do. We mention that our methods extend with minor changes to ℓ_p regression for $p > 1$. This is discussed in section 5.
- A natural extension of our algorithm to matrix-valued RHSs b gives a $(1+\varepsilon)$ approximation in a similar running time for the ℓ_1 norm subspace approximation problem. See section 4.3 for details.
- We can further improve the efficiency of solving this simple ℓ_1 regression problem, thereby replacing the $nd\log n$ running time term in Theorem 5 with $nd\log(d\varepsilon^{-1}\log n)$, but at the expense of a slightly larger sample size s. The improved algorithm is essentially the same as the FastCauchyRegression algorithm, except with two differences: Π_2 is chosen to be a matrix of i.i.d. Gaussians for a value $r_2 = O(\log(d\varepsilon^{-1}\log n))$; and, to accommodate this, the size of s needs to be increased. Details are presented in section 4.2.2.

4.2.2. A faster algorithm for ℓ_1 regression. Here, we present an algorithm that improves the efficiency of our ℓ_1 regression algorithm from section 4.2.1; and we state and prove an associated quality-of-approximation theorem. See Figure 3, which presents the OptimizedFastCauchyRegression algorithm. This algorithm has a somewhat larger sample size s than our previous algorithm, but our main theorem for this algorithm will replace the $nd\log n$ running time term in Theorem 5 with a $nd\log(d\varepsilon^{-1}\log n)$ term.

The intuition behind the OptimizedFastCauchyRegression algorithm is as follows. The (i,j)th entry $(U\Pi_2)_{ij}$ will be a 0-mean Gaussian with variance $\|U_{(i)}\|_2^2$. Since the row has d dimensions, the ℓ_2 norm and ℓ_1 norm only differ by $\sqrt{d}$. Hence, at the expense of some factors of d in the sampling complexity s, we can use sampling probabilities based on the ℓ_2 norms. The nice thing about using ℓ_2 norms is that we can use Gaussian random variables for the entries of Π_2 rather than Cauchy random variables. Given the exponential tail of a Gaussian random variable, for a Π_2 with

OptimizedFastCauchyRegression(A, b):

1: Let $X = \begin{bmatrix} A & -b \end{bmatrix} \in \mathbb{R}^{n \times (d+k)}$, and construct Π_1, an $r_1 \times n$ matrix satisfying (7) with A replaced by X.
2: Compute $X' = \Pi_1 X \in \mathbb{R}^{r_1 \times (d+k)}$ and its QR factorization, $\Pi_1 X = QR$. (Note that $\Pi_1 X R^{-1}$ has orthonormal columns.)
3: Set the parameters

$$s = \frac{210\kappa^2\sqrt{r_1(d+k)}}{\varepsilon^2}\left((d+k)\log\frac{24\kappa(d+k)\sqrt{r_1}}{\varepsilon} + \log\frac{2}{\delta}\right),$$
$$r_2 = 2\log\left(2sq\sqrt{r_1}\log^{2\rho+1/2} n\right) = O\left(\log\left(\rho(d+k)\epsilon^{-1}\log n\right)\right).$$

4: Let $\Pi_2 \in \mathbb{R}^{(d+k)\times r_2}$ be a matrix of independent standard Gaussians.
5: Construct $\Lambda = XR^{-1}\Pi_2 \in \mathbb{R}^{n \times r_2}$.
6: For $i \in [n]$, compute $\hat{\lambda}_i = \text{median}_{j\in[r_2]} |\Lambda_{ij}|$.
7: For $i \in [n]$, compute probabilities $\hat{p}_i = \min\{1, s \cdot \hat{\lambda}_i\}$.
8: Let $D \in \mathbb{R}^{n\times n}$ be diagonal with independent entries: $D_{ii} = \begin{cases} \frac{1}{\hat{p}_i}, & \text{prob. } \hat{p}_i; \\ 0, & \text{prob. } 1-\hat{p}_i. \end{cases}$
9: Return $\hat{x} \in \mathbb{R}^d$ that minimizes $\|DAx - Db\|_1$ with respect to x (using linear programming).

FIG. 3. *An optimized version of our main algorithm for solving ℓ_1 regression. Note that for this algorithm, Π_2 consists of independent Gaussian random variables and achieves the desired running time at the cost of a larger corset size, increased by a factor of* $\text{poly}(d\epsilon^{-1}\log n)$.

fewer columns we can still guarantee that no sampling probability *increases* by more than a logarithmic factor. The main difficulty we encounter is that some sampling probabilities may *decrease* by a larger factor, even though they do not increase by much—however, one can argue that with large enough probability, no row is sampled by the algorithm if its probability shrinks by a large factor. Therefore, the behavior of the algorithm is as if all sampling probabilities change by at most a $\text{poly}(d\varepsilon^{-1}\ln n)$ factor, and the result will follow. The following theorem is our main theorem for the OptimizedFastCauchyRegression algorithm. The proof of this theorem may be found in Appendix D.

THEOREM 6 (optimized Fast Cauchy ℓ_1 regression). *Given are $\varepsilon \in (0,1)$, $\rho > 0$, $A \in \mathbb{R}^{n\times d}$, and $b \in \mathbb{R}^n$. OptimizedFastCauchyRegression(A, b) constructs a coreset specified by the diagonal sampling matrix D and a solution vector $\hat{x} \in \mathbb{R}^d$ that minimizes the weighted regression objective $\|D(Ax-b)\|_1$. The solution $\hat{x}$ satisfies, with probability at least $1 - \frac{1}{d^\rho} - \frac{1}{\log^\rho n}$,*

$$\|A\hat{x} - b\|_1 \le \left(\frac{1+\varepsilon}{1-\varepsilon}\right)\|Ax - b\|_1 \qquad \forall x \in \mathbb{R}^d.$$

Further, with probability $1 - o(1)$, the entire algorithm to construct $\hat{x}$ runs in time

$$O\left(nd\log(\rho d\varepsilon^{-1}\log n) + \phi(s,d)\right) = O\left(nd\log(\rho d\varepsilon^{-1}\log n) + \text{poly}(d, \log(d\varepsilon^{-1}\ln n))\right),$$

where $\phi(s,d)$ is the time to solve an ℓ_1 regression problem on s vectors in d dimensions, and if FCT2 is used to construct Π_1, then $s = O\big(\frac{1}{\varepsilon^2} d^{2\rho+6+\eta}\log^{\frac{5}{2}}(\frac{d}{\varepsilon})\big)$.

Note that our algorithms and results also extend to multiple regression with $b \in \mathbb{R}^{n\times k}$, a fact that will be exploited in the next section.

4.3. ℓ_1 norm subspace approximation. Finally, we consider the ℓ_1 *norm subspace approximation problem*: Given the n points in the $n \times d$ matrix A and a parameter $k \in [d-1]$, embed these points into a subspace of dimension k to obtain the embedded points $\hat{A}$ such that $\|A - \hat{A}\|_1$ is minimized. (Note that this is the ℓ_1 analogue of the ℓ_2 problem that is solved by the SVD.) When $k = d-1$, the subspace is a hyperplane, and the task is to find the hyperplane passing through the origin so as to minimize the sum of ℓ_1 distances of the points to the hyperplane. In order to solve this problem with the methods from section 4.2, we take advantage of the observation made in [5] (see also Lemma 18 of [24]) that this problem can be reduced to d related ℓ_1 regressions of A onto each of its columns, a problem sometimes called *multiple regression*. Thus, in section 4.3.1, we extend our ℓ_1 "simple" regression algorithm to an ℓ_1 "multiple" regression algorithm; and then in section 4.3.2, we show how this can be used to solve the ℓ_1 norm subspace approximation problem.

4.3.1. Generalizing to multiple ℓ_1 regression. The multiple ℓ_1 regression problem is similar to the simple ℓ_1 regression problem, except that it involves solving for multiple RHSs; i.e., both x and b become matrices (W and B, respectively). Specifically, let $A \in \mathbb{R}^{n\times d}$ and $B \in \mathbb{R}^{n\times k}$. We wish to find $W \in \mathbb{R}^{d\times k}$ which solves

$$\min_W \|AW - B\|_1.$$

Although the optimal W can clearly be obtained by solving k separate simple ℓ_1 regressions, with $b = B^{(j)}$ for $j \in [k]$, one can do better. As with simple regression, we can reformulate the more general constrained optimization problem:

$$\min_{Z\in\mathcal{C}} \|XZ\|_1.$$

To recover multiple ℓ_1 regression, we set $X = \begin{bmatrix} A & -B \end{bmatrix}$ and $Z^T = \begin{bmatrix} W & I_k \end{bmatrix}^T$, in which case the constraint set is $\mathcal{C} = \left\{ Z = \begin{bmatrix} W \\ I_k \end{bmatrix} : W \in \mathbb{R}^{d\times k} \right\}$.

A detailed inspection of the proof of Theorem 5 in section 4.2 (see Appendix C for the proof) reveals that nowhere is it necessary that x be a vector, i.e., the whole proof generalizes to a matrix Z. In particular, the inequalities in (9) continue to hold since if they hold for every vector x, then it must hold for a matrix Z because $\|XZ\|_1 = \sum_{j\in[k]} \|XZ^{(j)}\|_1$. Similarly, if Lemma 13 continues to hold for *vectors*, then it will imply the desired result for matrices, and so the only change in all the algorithms and results is that the short dimension of X changes from $d+1$ to $d+k$. Thus, by shrinking δ by an additional factor of k and taking a union bound, we get a relative error approximation for each individual regression. We refer to this modified algorithm, where a matrix B is input and the optimization problem in the last step is modified appropriately, as FastCauchyRegression(A, B), overloading notation in the obvious way. This discussion is summarized in the following theorem.

THEOREM 7 (Fast Cauchy multiple ℓ_1 regression). *Given $\varepsilon \in (0,1)$, $\rho > 0$, a matrix $A \in \mathbb{R}^{n\times d}$, and $B \in \mathbb{R}^{n\times k}$, FastCauchyRegression$(A, B)$ constructs a coreset specified by the diagonal sampling matrix D and a solution $\hat{W} \in \mathbb{R}^{d\times k}$ that minimizes the weighted multiple regression objective $\|D(AW - B)\|_1$. The solution $\hat{W}$ satisfies, with probability at least $1 - \frac{1}{(d+k)^\rho}$,*

$$\|A\hat{W}^{(j)} - B^{(j)}\|_1 \le \left(\frac{1+\varepsilon}{1-\varepsilon}\right) \|Ax - B^{(j)}\|_1 \qquad \forall x \in \mathbb{R}^d \text{ and } \forall j \in [k].$$

Further, with probability $1 - o(1)$*, the entire algorithm to construct* $\hat{W}$ *runs in time*

$$O\left(n(d+k)\log n + \phi(s,d,k)\right),$$

where $\phi(s,d,k)$ *is the time to solve* k ℓ_1 *regression problems on the same* s *vectors in* d *dimensions, and if FCT2 is used to construct* Π_1*, then*

$$s = O\left(\frac{1}{\varepsilon^2}(d+k)^{\rho+\frac{11}{2}+\eta}\log^{\frac{3}{2}}\left(\frac{d+k}{\varepsilon}\right)\right).$$

Remarks. Several remarks about our results for this ℓ_1 multiple regression problem are in order:

- First, we can save an extra factor of $(d+k)$ in s in the above theorem if all we want is a relative error approximation to the entire multiple regression and we do not need relative error approximations to each individual regression.
- Second, when $k = O(d)$ it is interesting that there is essentially no asymptotic overhead in solving this problem other than the increase from $\phi(s,d)$ to $\phi(s,d,k)$; in general, by preprocessing the matrix DA, solving k regressions on this same matrix DA is much quicker than solving k separate regressions. This should be compared with ℓ_2 regression, where solving k regressions with the same A takes $O(nd^2 + ndk + kd^2)$ (since the SVD of A needs to be done only once), versus a time of $O(nkd^2)$ for k separate ℓ_2 regressions.
- Third, we will use this version of ℓ_1 multiple regression problem, which is more efficient than solving k separate ℓ_1-simple regressions, to solve the ℓ_1-subspace approximation problem. See section 4.3.2 for details.

4.3.2. Application to ℓ_1 norm subspace approximation. Here, we will take advantage of the observation made in [5] that the ℓ_1 norm subspace approximation problem can be reduced to d related ℓ_1 regressions of A onto each of its columns. To see this, consider the following ℓ_1 regression problem:

$$\min_{w:w_j=0} \|Aw - A^{(j)}\|_1.$$

This regression problem is fitting (in the ℓ_1 norm) the jth column of A onto the remaining columns. Let w_j^* be an optimal solution. Then if we replace $A^{(j)}$ by Aw_j^*, the resulting vectors will all be in a $(d-1)$-dimensional subspace. Let A_j be A with $A^{(j)}$ replaced by Aw_j^*. The crucial observation made in [5] (see also Lemma 18 of [24]) is that *one of the* A_j *is optimal*—and so the optimal subspace can be obtained by simply doing a hyperplane fit to the embedded points. So,

$$\min_{j\in[d]} \|A - A_j\|_1 = \min_{\mathbf{rank}(\hat{A})=d-1} \|A - \hat{A}\|_1.$$

When viewed from this perspective, the ℓ_1 norm subspace approximation problem makes the connection between low-rank matrix approximation and overconstrained ℓ_1 regression. (A similar approach was used in the ℓ_2 case to obtain relative-error low-rank CX and CUR matrix decompositions [11, 21].) We thus need to perform k constrained regressions, which can be formulated into a single constrained multiple regression problem, which can be solved as follows: Find the matrix W that solves

$$\min_{W\in\mathcal{C}} \|AW\|_1,$$

where the constraint set is $\mathcal{C} = \{W \in \mathbb{R}^{d\times d} : W_{ii} = -1\}$. Since the constraint set effectively places an independent constraint on each column of W, after some elementary manipulation, it is easy to see that this regression is equivalent to the d individual regressions to obtain w_j^*. Indeed, for an optimal solution W^*, we can set $w_j^* = W^{*(j)}$.

Thus, using our approximation algorithm for constrained multiple ℓ_1 regression that we described in section 4.3.1, we can build an approximation algorithm for the ℓ_1 norm subspace approximation problem that improves upon the previous best algorithm from [24, 5]. (The running time of the algorithm of [24] is $\Omega(nd^{\omega^+} + \mathrm{poly}(d\varepsilon^{-1}\log n))$, where $\omega \approx 2.376$ and $\beta > 0$ is any constant.) Our improved algorithm is basically our multiple ℓ_1 regression algorithm, $\mathsf{FastCauchyRegression}(A, B)$, invoked with A and $b = \{\}$ (NULL). The algorithm proceeds exactly as outlined in Figure 2, except for the last step, which instead uses linear programming to solve for $\hat{W}$ that minimizes $\|AW\|_1$ with respect to $W \in \mathcal{C}$. (Note that the constraints defining $\mathcal{C}$ are very simple affine equality constraints.) Given $\hat{W}$, we define $\hat{w}_j = \hat{W}^{(j)}$ and compute $j^* = \mathrm{argmin}_{j\in[d]} \|A - \hat{A}_j\|$, where $\hat{A}_j$ is A with the column $A^{(j)}$ replaced by $A\hat{w}_j$. It is easy to now show that $\hat{A}_{j^*}$ is a $(1+\varepsilon)$-approximation to the $(d-1)$-dimensional subspace approximation problem. Indeed, recall that W^* is optimal and the optimal error is $\|AW^{*(j)}\|_1$ for some $j \in [d]$; however, for any $j \in [d]$,

$$\|AW^{*(j)}\|_1 \overset{(a)}{\geq} \left(\frac{1-\varepsilon}{1+\varepsilon}\right) \|A\hat{W}^{(j)}\|_1 \overset{(b)}{\geq} \left(\frac{1-\varepsilon}{1+\varepsilon}\right) \|A\hat{W}^{(j^*)}\|_1 = \left(\frac{1-\varepsilon}{1+\varepsilon}\right) \|A - \hat{A}_{j^*}\|_1,$$

where (a) is from the $(1+\varepsilon)$-optimality of the constrained multiple regression as analyzed in Appendix C and (b) is because j^* attained minimum error among all $j \in [d]$. This discussion is summarized in the following theorem.

THEOREM 8. *Given $A \in \mathbb{R}^{n\times d}$ (n points in d dimensions), there is a randomized algorithm which outputs a $(1+\varepsilon)$-approximation to the ℓ_1 norm subspace approximation problem for these n points with probability at least $1 - \frac{1}{d^\rho}$. Further, the running time, with probability $1 - o(1)$, is*

$$O\left(nd\log n + \tfrac{1}{\varepsilon^2}\,\mathrm{poly}(d, \log\tfrac{d}{\varepsilon})\right).$$

5. Extensions to ℓ_p for $p > 1$. In this section, we describe extensions of our methods to ℓ_p for $p > 1$. We will first (in section 5.1) discuss ℓ_p norm conditioning and connect it to ellipsoidal rounding, followed by a fast rounding algorithm for general centrally symmetric convex sets (in section 5.2); and we will then (in section 5.3) show how to obtain quickly a well-conditioned basis for the ℓ_p norm for any $p \in [1, \infty)$ and (in section 5.4) show how this basis can be used for improved ℓ_p regression. These results will generalize our results for ℓ_1 from sections 4.1 and 4.2, respectively, to general ℓ_p.

5.1. ℓ_p norm conditioning and ellipsoidal rounding. As with ℓ_2 regression, ℓ_p regression problems are easier to solve when they are well-conditioned. Thus, we start with the definition of the ℓ_p norm condition number κ_p of a matrix A.

DEFINITION 5 (ℓ_p norm conditioning). *Given an $n \times d$ matrix A, let*

$$\sigma_p^{\max}(A) = \max_{\|x\|_2\leq 1} \|Ax\|_p \quad \text{and} \quad \sigma_p^{\min}(A) = \min_{\|x\|_2\geq 1} \|Ax\|_p.$$

Then, we denote by $\kappa_p(A)$ the ℓ_p norm condition number of A, defined to be

$$\kappa_p(A) = \sigma_p^{\max}(A)/\sigma_p^{\min}(A).$$

For simplicity, we will use κ_p, $\sigma_p^{\min}$, and $\sigma_p^{\max}$ when the underlying matrix is clear.

There is a strong connection between the ℓ_p norm condition number and the concept of an (α, β, p)-conditioning developed by Dasgupta et al. [7].

DEFINITION 6 ((α, β, p)-conditioning (from [7])). *Given an $n \times d$ matrix A and $p \in [1, \infty]$, let $\|\cdot\|_q$ be the dual norm of $\|\cdot\|_p$, i.e., $1/p + 1/q = 1$. Then, A is* (α, β, p)-conditioned *if* (1) $\|A\|_p \leq \alpha$, *and* (2) *for all $z \in \mathbb{R}^d$, $\|z\|_q \leq \beta \|Az\|_p$. Define $\bar{\kappa}_p(A)$ as the minimum value of $\alpha\beta$ such that A is (α, β, p)-conditioned. We say that A is* ℓ_p well-conditioned *if $\bar{\kappa}_p(A) = \mathcal{O}(\text{poly}(d))$, independent of n.*

The following lemma characterizes the relationship between these two quantities.

LEMMA 10. *Given an $n \times d$ matrix A and $p \in [1, \infty]$, we always have*

$$d^{-|1/2-1/p|} \kappa_p(A) \leq \bar{\kappa}_p(A) \leq d^{\max\{1/2, 1/p\}} \kappa_p(A).$$

Proof. To see the connection, recall that

$$\|A\|_p = \left(\sum_{j=1}^{d} \|Ae_j\|_p^p \right)^{1/p} \leq \left(\sum_{j=1}^{d} (\sigma_p^{\max} \|e_j\|_2)^p \right)^{1/p} = d^{1/p} \sigma_p^{\max}$$

and that

$$\|Ax\|_p \geq \sigma_p^{\min} \|x\|_2 \geq d^{\min\{1/p-1/2, 0\}} \sigma_p^{\min} \|x\|_q \quad \forall x \in \mathbb{R}^n.$$

Thus, A is $(d^{1/p}\sigma_p^{\max}, 1/(d^{\min\{1/p-1/2,0\}}\sigma_p^{\min}), p)$-conditioned and

$$\bar{\kappa}_p(A) \leq d^{\max\{1/2, 1/p\}} \kappa_p(A).$$

On the other hand, if A is (α, β, p)-conditioned, we have, for all $x \in \mathbb{R}^d$,

$$\|Ax\|_p \leq \|A\|_p \|x\|_q \leq d^{\max\{1/2-1/p, 0\}} \alpha \cdot \|x\|_2$$

and

$$\|Ax\|_p \geq \|x\|_q / \beta \geq d^{\min\{1/2-1/p, 0\}} / \beta \cdot \|x\|_2.$$

Thus, $\kappa_p(A) \leq d^{|1/p-1/2|}\alpha\beta$. ∎

Although it is easier to describe sampling algorithms in terms of $\bar{\kappa}_p$, after we show the equivalence between κ_p and $\bar{\kappa}_p$, it will be easier for us to discuss conditioning algorithms in terms of κ_p, which naturally connects to ellipsoidal rounding algorithms.

DEFINITION 7. *Let $\mathcal{C} \subseteq \mathbb{R}^d$ be a convex set that is full-dimensional, closed, bounded, and centrally symmetric with respect to the origin. An ellipsoid $\mathcal{E} = \{x \in \mathbb{R}^d \,|\, \|Rx\|_2 \leq 1\}$ is a* κ-rounding *of $\mathcal{C}$ if it satisfies $\mathcal{E}/\kappa \subseteq \mathcal{C} \subseteq \mathcal{E}$ for some $\kappa \geq 1$, where $\mathcal{E}/\kappa$ means shrinking $\mathcal{E}$ by a factor of $1/\kappa$.*

To see the connection between rounding and conditioning, let

$$\mathcal{C} = \{x \in \mathbb{R}^d \,|\, \|Ax\|_p \leq 1\}$$

and assume that we have a κ-rounding of $\mathcal{C}$: $\mathcal{E} = \{x \,|\, \|Rx\|_2 \leq 1\}$. This implies

$$\|Rx\|_2 \leq \|Ax\|_p \leq \kappa \|Rx\|_2 \quad \forall x \in \mathbb{R}^d.$$

If we let $y = Rx$, then we get

$$\|y\|_2 \leq \|AR^{-1}y\|_p \leq \kappa \|y\|_2 \quad \forall y \in \mathbb{R}^d.$$

Therefore, we have $\kappa_p(AR^{-1}) \leq \kappa$. So a κ-rounding of $\mathcal{C}$ leads to a κ-conditioning of A.

5.2. Fast ellipsoidal rounding. Here, we provide a deterministic algorithm to compute a $2d$-rounding of a centrally symmetric convex set in $\mathbb{R}^d$ that is described by a separation oracle. Recall the well-known result due to John [17] that for a centrally symmetric convex set $\mathcal{C}$ there exists a $d^{1/2}$-rounding and that such a rounding is given by the Löwner–John (LJ) ellipsoid of $\mathcal{C}$, i.e., the minimal-volume ellipsoid containing $\mathcal{C}$. However, finding this $d^{1/2}$-rounding is a hard problem. To state algorithmic results, suppose that $\mathcal{C}$ is described by a separation oracle and that we are provided an ellipsoid $\mathcal{E}_0$ that gives an L-rounding for some $L \geq 1$. In this case, the best known algorithmic result of which we are aware is that we can find a $(d(d+1))^{1/2}$-rounding in polynomial time, in particular, in $O(d^4 \log L)$ calls to the oracle; see Lovász [19, Theorem 2.4.1]. This result was used by Clarkson [6] and by Dasgupta et al. [7]. Here, we follow the same construction, but we show that it is much faster to find a (slightly worse) $2d$-rounding. The proof of this theorem may be found in Appendix E.1.

THEOREM 9 (fast ellipsoidal rounding). *Given a centrally symmetric convex set $\mathcal{C} \subseteq \mathbb{R}^d$ centered at the origin and described by a separation oracle, and an ellipsoid $\mathcal{E}_0$ centered at the origin such that $\mathcal{E}_0/L \subseteq \mathcal{C} \subseteq \mathcal{E}_0$ for some $L \geq 1$, it takes at most $3.15d^2 \log L$ calls to the oracle and additional $O(d^4 \log L)$ time to find a $2d$-rounding of $\mathcal{C}$.*

Applying Theorem 9 to the convex set $\mathcal{C} = \{x \,|\, \|Ax\|_p \leq 1\}$, with the separation oracle described via a subgradient of $\|Ax\|_p$ and the initial rounding provided by the "R" matrix from the QR decomposition of A, we improve the running time of the algorithm used by Clarkson [6] and by Dasgupta et al. [7] from $\mathcal{O}(nd^5 \log n)$ to $\mathcal{O}(nd^3 \log n)$ while maintaining an $\mathcal{O}(d)$-conditioning. The proof of this theorem may be found in Appendix E.2.

THEOREM 10. *Given an $n \times d$ matrix A with full column rank, it takes at most $\mathcal{O}(nd^3 \log n)$ time to find a matrix $R \in \mathbb{R}^{d \times d}$ such that $\kappa_p(AR^{-1}) \leq 2d$.*

5.3. Fast construction of an ℓ_p well-conditioned basis. Here, we consider the construction of a basis that is well-conditioned for ℓ_p. To obtain results for general ℓ_p that are analogous to those we obtained for ℓ_1, we will extend the FCT2 construction from section 3.2, combined with Theorem 9.

Our main algorithm for constructing a p-well-conditioned basis, the FastLpBasis algorithm, is summarized in Figure 4. The algorithm first applies blockwise embeddings in the ℓ_2 norm, similar to the construction of FCT2; it then uses the algorithm of Theorem 9 to compute a $(2d)$-rounding of a special convex set $\tilde{\mathcal{C}}$ and obtain the matrix R. It is thus a generalization of our FastL1Basis algorithm of section 4.1, and it follows the same high-level structure laid out by the algorithm of [10] for computing approximations to the ℓ_2 leverage scores and an ℓ_2 well-conditioned basis.

The next theorem is our main result for the FastLpBasis algorithm. It improves the running time of the algorithm of Theorem 10, at the cost of slightly worse conditioning quality. However, these worse factors will only contribute to a low-order additive poly(d) term in the running time of our ℓ_p regression application in section 5.4. The proof of this theorem may be found in Appendix F.

THEOREM 11 (fast ℓ_p well-conditioned basis). *For any $A \in \mathbb{R}^{n \times d}$ with full column rank, the basis AR^{-1} constructed by FastLpBasis(A) (Figure 4), with probability at least $1 - 1/n$, is ℓ_p well-conditioned with $\kappa_p(AR^{-1}) = \mathcal{O}(dt^{|1/p-1/2|})$. The time to*

FastLpBasis(A):

1: Let $s = \Theta(d + \log n)$, let $t = \Theta(sd^2)$, and let G be an $s \times t$ Fast Johnson–Lindenstrauss matrix, the same as the matrix G in the FCT2 construction.

2: Partition A along its rows into submatrices of size $t \times d$, denoted by $A_1, \ldots, A_N$, compute $\tilde{A}_i = GA_i$ for $i = 1, \ldots, N$, and define

$$\tilde{\mathcal{C}} = \left\{ x \;\middle|\; \left(\sum_{i=1}^{N} \|\tilde{A}_i x\|_2^p \right)^{1/p} \leq 1 \right\} \text{ and } \tilde{A} = \begin{pmatrix} \tilde{A}_1 \\ \vdots \\ \tilde{A}_N \end{pmatrix}.$$

3: Apply the algorithm of Theorem 9 to obtain a $(2d)$-rounding of $\tilde{\mathcal{C}}$: $\mathcal{E} = \{x \mid \|Rx\|_2 \leq 1\}$.

4: Output AR^{-1}.

FIG. 4. *Our main algorithm for the fast construction of an ℓ_p well-conditioned basis of an $n \times d$ matrix A. Note the structural similarities with our* FastL1Basis *algorithm of Figure 1 for computing quickly an ℓ_1 well-conditioned basis.*

compute R is $O(nd \log n)$.

When $d > \log n$, $\kappa_p(AR^{-1}) = \mathcal{O}(d^{1+3\cdot|1/p-1/2|})$ and hence

$$\bar{\kappa}_p(AR^{-1}) = \mathcal{O}(d^{1+3\cdot|1/p-1/2|+\max\{1/p,1/2\}})$$

by Lemma 10. Note that, even for the case when $p = 1$, we have $\bar{\kappa}_p(AR^{-1}) = \mathcal{O}(d^{7/2})$, which is slightly better than FCT2 (see Corollary 1). However, we have to solve a rounding problem of size $ns/t \times d$ in step 2 of FastLpBasis, which requires storage and work depending on n.

5.4. Fast ℓ_p regression. Here, we show that the overconstrained ℓ_p regression problem can be solved with a generalization of the algorithms of section 4.2 for solving ℓ_1 regression; we will call this generalization the FastLpRegression algorithm. In particular, as with the algorithm for ℓ_1 regression, this FastLpRegression algorithm for the ℓ_p regression problem uses an ℓ_p well-conditioned basis and samples rows of A with probabilities proportional to the ℓ_p norms of the rows of the corresponding well-conditioned basis (which are the ℓ_p analogues of the ℓ_2 leverage scores). As with the FastCauchyRegression, this entails using—for speed—a second random projection Π_2 applied to AR^{-1}—on the right—to estimate the row norms. This allows fast estimation of the ℓ_2 norms of the rows of AR^{-1}, which provides an estimate of the ℓ_p norms of those rows, up to a factor of $d^{|1/2-1/p|}$. We use these norm estimates, e.g., as in the above algorithms or in the sampling algorithm of [7]. As discussed for the running time bound of [7, Theorem 7], this algorithm samples a number of rows proportional to $\bar{\kappa}_p^p(AR^{-1})d$. This factor, together with a sample complexity increase of $(d^{|1/2-1/p|})^p = d^{|p/2-1|}$ needed to compensate for error due to using Π_2, gives a sample complexity increase for the FastLpRegression algorithm, while the leading term in the complexity (for $n \gg d$) is reduced from $O(nd^5 \log n)$ to $O(nd \log n)$. We modify Theorem 7 of [7] to obtain the following theorem.

THEOREM 12 (fast ℓ_p regression). *Given $\varepsilon \in (0, 1)$, $A \in \mathbb{R}^{n\times d}$, and $b \in \mathbb{R}^n$, there is a random sampling algorithm (the* FastLpRegression *algorithm described above) for ℓ_p regression that constructs a coreset specified by a diagonal sampling matrix D and a solution vector $\hat{x} \in \mathbb{R}^d$ that minimizes the weighted regression objective $\|D(Ax - b)\|_p$.*

The solution $\hat{x}$ satisfies, with probability at least $1/2$, the relative error bound that $\|A\hat{x} - b\|_p \leq (1+\varepsilon)\|Ax - b\|_p$ for all $x \in \mathbb{R}^d$. Further, with probability $1 - o(1)$, the entire algorithm to construct $\hat{x}$ runs in time

$$O\left(nd \log n + \phi_p(s,d)\right) = O\left(nd \log n + \tfrac{1}{\varepsilon^2} \operatorname{poly}(d, \log \tfrac{d}{\varepsilon})\right),$$

where $s = O(\varepsilon^{-2} d^k \log(1/\varepsilon))$ with $k = p + 1 + 4|p/2 - 1| + \max\{p/2, 1\}$, and $\phi_p(s,d)$ is the time to solve an ℓ_p regression problem on s vectors in d dimensions.

6. Numerical implementation and empirical evaluation. In this section, we describe the results of our empirical evaluation. We have implemented and evaluated the Fast Cauchy transforms (both FCT1 and FCT2) as well as the Cauchy transform (CT) of [24]. For completeness, we have also compared our method against two ℓ_2-based transforms: the Gaussian transform (GT) and a version of the FJLT. Ideally, the evaluation would be based on evaluating the distortion of the embedding, i.e., evaluating the smallest κ such that

$$\|Ax\|_1 \leq \|\Pi Ax\|_1 \leq \kappa \|Ax\|_1 \quad \forall x \in \mathbb{R}^d,$$

where $\Pi \in \mathbb{R}^{r \times n}$ is one of the CTs. Due to the nonconvexity, there seems not to be a way to compute, tractably and accurately, the value of this κ. Instead, we evaluate both ℓ_1-based transforms (CT, FCT1, and FCT2) and ℓ_2-based transforms (GT and FJLT) based on how they perform in computing well-conditioned bases and approximating ℓ_1 regression problems.

6.1. Evaluating the quality of ℓ_1 well-conditioned bases. We first describe our methodology. Given a "tall and skinny" matrix $A \in \mathbb{R}^{n \times d}$ with full column rank, as in section 4.1, we compute well-conditioned bases of A: $U = AR^{-1} = A(Q^T \Pi A)^{-1}$, where Π is one of those transforms, and where Q and R are from the QR decomposition of ΠA. Our empirical evaluation is based on the metric $\bar{\kappa}_1(U)$. Note that $\bar{\kappa}_1$ is scale-invariant: if U is (α, β)-conditioned with $\bar{\kappa}_1(U) = \alpha\beta$, then γU is $(\alpha\gamma, \beta/\gamma)$-conditioned, and hence $\bar{\kappa}_1(\gamma U) = \alpha\gamma\beta/\gamma = \alpha\beta = \bar{\kappa}_1(U)$. This saves us from determining the scaling constants when implementing CT, FCT1, and FCT2. While computing $\alpha = \|U\|_1$ is trivial, computing $\beta = 1/(\min_{\|z\|_\infty = 1} \|Uz\|_1)$ is not as easy: it requires solving d linear programs:

$$\beta = \frac{1}{\min\limits_{j=1,\ldots,d} \; \min\limits_{\substack{\|z\|_\infty \leq 1 \\ z_j = 1}} \|Uz\|_1}.$$

Note that this essentially limits the size of the test problems in our empirical evaluation: although we have applied our algorithms to much larger problems, we must solve these linear programs if we want to provide a meaningful comparison by comparing our fast ℓ_1-based algorithms with an "exact" answer. Another factor limiting the size of our test problems is more subtle and is a motivation for our comparison with ℓ_2-based algorithms. Consider a basis induced by the GT: $U = A(Q^T GA)^{-1}$, where $G \in \mathbb{R}^{O(d) \times n}$ is a matrix whose entries are i.i.d. Gaussian. We know that $\kappa_2(U) = O(1)$ with high probability. In such a case, we have

$$\|U\|_1 = \sum_{j=1}^{d} \|Ue_j\|_1 \leq \sum_{j=1}^{d} n^{1/2} \|Ue_j\|_2 \leq n^{1/2} d \cdot \sigma_2^{\max}(U)$$

TABLE 1
Summary of time complexity and ℓ_1 conditioning performance for ℓ_1-based and ℓ_2-based transforms used in our empirical evaluation.

	Time	$\bar{\kappa}_1$
CT	$O(nd^2 \log d)$	$O(d^{5/2} \log^{3/2} d)$
FCT1	$O(nd \log d)$	$O(d^{11/2} \log^{9/2} d)$
FCT2	$O(nd \log d)$	$O(d^{7/2+\eta} \log^{3/2} d)$
GT	$O(nd^2)$	$O(n^{1/2}d)$
FJLT	$O(nd \log n)$	$O(n^{1/2}d)$

and

$$\|Uz\|_1 \geq \|Uz\|_2 \geq \sigma_2^{\min}(U)\|z\|_2 \geq \sigma_2^{\min}(U)\|z\|_\infty.$$

Hence $\bar{\kappa}_1(U) \leq n^{1/2}d \cdot \sigma_2^{\max}(U)/\sigma_2^{\min}(U) = O(n^{1/2}d)$. Similar results apply to the FJLTs that work on an entire subspace of vectors, e.g., the subsampled randomized Hadamard transform (SRHT) [28]. In our empirical evaluation, we use the SRHT as our implementation of the FJLT, but we note that similar running times hold for other variants of the FJLT [4]. Table 1 lists the running time and worst-case performance of each transform on ℓ_1 conditioning, clearly showing the cost-performance trade-offs. For example, comparing the condition number of GT or FJLT, $O(n^{1/2}d)$, with the condition number of CT, $O(d^{5/2} \log^{3/2} d)$, we will need $n > O(d^3 \log^3 d)$ to see the advantage of CT over ℓ_2-based algorithms (e.g., n should be at least at the scale of 10^8 when d is 100). To observe the advantage of FCT1 and FCT2 over ℓ_2-based transforms, n should be relatively even larger.

Motivated by these observations, we create two sets of test problems. The first set contains matrices of size $2^{18} \times 4$, and the second set contains matrices of size $2^{16} \times 16$. We choose the number of rows to be powers of 2 to implement FCT2 and FJLT in a straightforward way. Based on our theoretical analysis, we expect ℓ_1-based algorithms to work better on the first test set than ℓ_2-based algorithms, at least on some worst-case test problems, and that this advantage should disappear on the second test set. For each of these two sizes, we generate four test matrices: A_1 is a randomly generated ill-conditioned matrix with slightly heterogeneous leverage scores; A_2 is a randomly generated ill-conditioned matrix with strongly heterogeneous leverage scores; and A_3 and A_4 are two "real" matrices chosen to illustrate the performance of our algorithms on real-world data. In more detail, the test matrices are as follows:

- $A_1 = D_1 G_1 D_2 G_2$, where $D_1 \in \mathbb{R}^{n \times n}$ is a diagonal matrix whose diagonals are linearly spaced between 1 and 10^4, $G_1 \in \mathbb{R}^{n \times d}$ is a Gaussian matrix, $D_2 \in \mathbb{R}^{d \times d}$ is a diagonal matrix whose diagonals are linearly spaced between 1 and 10^4, and $G_2 \in \mathbb{R}^{d \times d}$ is a Gaussian matrix. A_1 is chosen in this way so that it is ill-conditioned (due to the choice of D_2) and its bottom rows tend to have high leverage scores (due to the choice of D_1).
- $$A_2 = \begin{pmatrix} 1 & & \\ & \ddots & \\ & & 1 \cdots 1 \end{pmatrix}^T G,$$
  where $G \in \mathbb{R}^{d \times d}$ is a Gaussian matrix. The first $d-1$ rows tend to have very high leverage scores because missing any of them would lead to rank deficiency, while the rest of the $n-d+1$ rows are the same as each other, and hence they tend to have very low leverage scores. A_2 is also ill-conditioned

TABLE 2

ℓ_1 norm conditioning, $\bar{\kappa}_1(U)$, on matrices of size $2^{18} \times 4$. We compute the first and the third quartiles of the ℓ_1 norm conditioning number in 50 independent runs for each matrix and each algorithm. The size is chosen to demonstrate the difference between ℓ_1-based and ℓ_2-based conditioning algorithms and the superiority of the ℓ_1-based algorithms in the asymptotic regime. GT and FJLT do not work well on A_2, resulting in condition numbers close to the worst-case bound of $n^{1/2}d = 2048$. CT, FCT1, and FCT2 perform consistently across all matrices.

	A_1	A_2	A_3	A_4
$\bar{\kappa}_1(A_i)$	1.93e+04	7.67e+05	8.58	112
CT	[10.8, 39.1]	[10.4, 41.7]	[10.2, 33]	[8.89, 42.8]
FCT1	[9.36, 21.2]	[15.4, 58.6]	[10.9, 38.9]	[11.3, 40.8]
FCT2	[12.3, 32.1]	[17.3, 76.1]	[10.9, 43]	[11.3, 42.1]
GT	[6.1, 8.81]	[855, 1.47e+03]	[5.89, 8.29]	[6.9, 9.17]
FJLT	[5.45, 6.29]	[658, 989]	[5.52, 6.62]	[6.18, 7.53]

because we have

$$A_2^T A_2 = G^T \begin{pmatrix} 1 & & \\ & \ddots & \\ & & 1 \\ & & & w \end{pmatrix} G,$$

where $w = (n - d + 1)^2$ is very large.

- A_3, the leading submatrix of the SNP matrix used by Paschou et al. [23]. The SNP matrix is of size 492516×2250, from the Human Genome Diversity Panel and the HapMap Phase 3 dataset. See [23] for more descriptions of the data.
- A_4, the leading submatrix of the TinyImages matrix created by Torralba, Fergus, and Freeman [26]. The original images are in RGB format. We convert them to grayscale intensity images, resulting in a matrix of size $8e7 \times 1024$.

To implement FCT1 and FCT2 for our empirical evaluations, we have to fix several parameters in Theorems 2 and 1, finding a compromise between theory and practice. We choose $r_1 = \lceil 2d \log d \rceil$ except $r_1 = 2d$ for GT. We choose $s = \lceil 2d \log d \rceil$ and $t = 2d^2$ for FCT1 and $s = 2^{\lceil 2 \log_2 (2d \log d) \rceil}$ for FCT2. Although those settings do not follow Theorems 2 and 1 very closely, they seem to be good for practical use. Since all the transforms are randomized algorithms that may fail with certain probabilities, for each test matrix and each transform, we take 50 independent runs and show the first and the third quartiles of $\bar{\kappa}_1$ in Tables 2 and 3.

The empirical results, described in detail in Tables 2 and 3, conform with our expectations. The specifically designed ℓ_1-based algorithms perform consistently across all test matrices, while the performance of ℓ_2-based algorithms is quite problem-dependent. Interestingly, though, the ℓ_2-based methods often perform reasonably well: at root, the reason is that for many inputs the ℓ_2 leverage scores are not too much different than the ℓ_1 leverage scores. That being said, the matrix A_2 clearly indicates that ℓ_2-based methods can fail for "worst-case" input, while the ℓ_1-based methods perform well for this input.

On the first test set, ℓ_1-based algorithms are comparable to ℓ_2-based algorithms on A_1, A_3, and A_4 but much better on A_2. The differences among ℓ_1-based algorithms are small. In terms of conditioning quality, CT leads FCT1 and FCT2 by a small amount on average, but when we take running times into account, FCT1 and FCT2 are clearly more favorable choices in this asymptotic regime. On the second test set, ℓ_1-based algorithms become worse than ℓ_2-based on A_1, A_3, and A_4 due to the increase of d and the decrease of n. All the algorithms perform similarly on A_2, but

TABLE 3

ℓ_1 norm conditioning, $\bar{\kappa}_1(U)$, on matrices of size $2^{16} \times 16$. We compute the first and the third quartiles of the ℓ_1 norm conditioning number in 50 independent runs for each matrix and each algorithm. The size is chosen to demonstrate that ℓ_2-based conditioning algorithms can be as good as or even better than ℓ_1-based conditioning algorithms. GT and FJLT still do not work well on A_2, but they become comparable to ℓ_1-based algorithms. Although still performing consistently across all matrices, ℓ_1-based algorithms perform much worse than in the first test set due to the increase of d and decrease of n.

	A_1	A_2	A_3	A_4
$\bar{\kappa}_1(A_i)$	4.21e+05	2.39e+06	36.5	484
CT	[90.2, 423]	[386, 1.44e+03]	[110, 633]	[150, 1e+03]
FCT1	[113, 473]	[198, 1.1e+03]	[114, 765]	[127, 684]
FCT2	[134, 585]	[237, 866]	[106, 429]	[104, 589]
GT	[27.4, 31]	[678, 959]	[28.8, 32.3]	[29.4, 33.5]
FJLT	[19.9, 21.2]	[403, 481]	[21.4, 23.1]	[21.8, 23.2]

ℓ_1-based algorithms, involving Cauchy random variables, have larger variance than ℓ_2-based algorithms.

6.2. Application to ℓ_1 regression. Next, we embed these transforms into fast approximation of ℓ_1 regression problems to see how they affect the accuracy of approximation. We implement the FastCauchyRegression algorithm of section 4.2, except that we compute the row norms of U exactly instead of estimating them. Although this takes $O(nd^2)$ time, it is free from errors introduced by estimating the row norms of U, and thus it permits a more direct evaluation of the regression algorithm. Unpublished results indicate that using approximations to the ℓ_1 leverage scores, as is done at the beginning of the FastCauchyRegression algorithm, leads to very similar quality-of-approximation results.

We generate a matrix A of size $2^{18} \times 7$ and generate the RHSs $b = Ax_{\text{exact}} + \varepsilon$, where x_{exact} is a Gaussian vector, and ε is a random vector whose entries are independently sampled from the Laplace distribution and scaled such that $\|\varepsilon\|_2/\|Ax_{\text{exact}}\|_2 = 0.1$. Then, for each row i, with probability 0.001 we replace b_i by $100\|\varepsilon\|_2$ to simulate corruption in measurements. On this kind of problem, ℓ_1 regression should give a very accurate estimate, while ℓ_2 regression will not work well. For completeness, we also add uniform sampling (UNIF) and no conditioning (NOCD) into the evaluation. Instead of determining the sample size from a given tolerance, we accept the sample size as a direct input; and we choose sample sizes from 2^5 to 2^{14}.

The results are shown in Figure 5, where we draw the first and the third quartiles of the relative errors in objective value from 50 independent runs. If the subsampled matrix is rank-deficient, we set the corresponding relative error to ∞ to indicate a failure. We remove relative errors that are larger than 100 from the plot in order to show more details. As expected, we can see that UNIF and NOCD are certainly not among reliable choices; they failed (either generating rank-deficient subsampled problems or relative errors larger than 100) completely on A_2. In addition, GT and FJLT failed partially on the same test. Empirically, there is not much difference among ℓ_1-based algorithms: CT works slightly worse than FCT1 and FCT2 on these tests, which certainly makes FCT1 and FCT2 more favorable. (One interesting observation is that we find that, in these tests at least, the relative error is proportional to $1/s$ instead of $1/s^{1/2}$. At this time, we do not have theory to support this observation.) This coupled with the fact that ℓ_1 leverage scores can be approximated more quickly

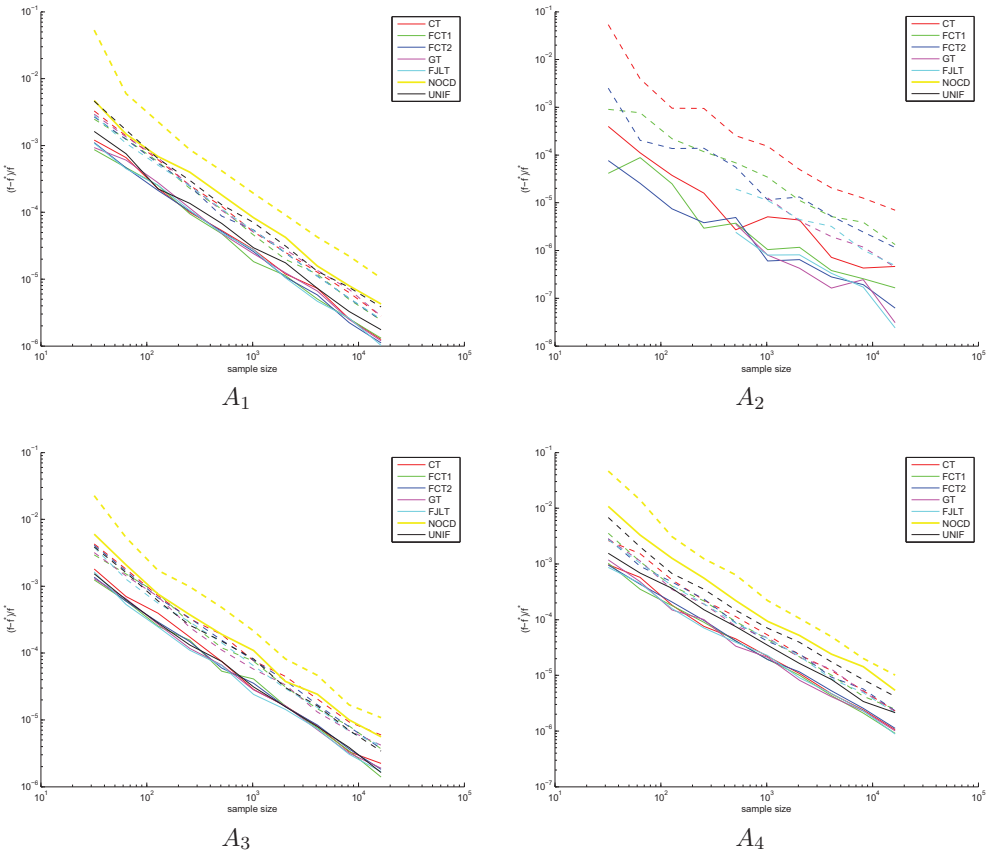

FIG. 5. *The first and the third quartiles of relative errors in objective value. The problem size is* $2^{18} \times 7$. *The first quartiles are drawn in solid lines, while the third quartiles are drawn in dashed lines. If the subsampled problem is rank-deficient, we set the corresponding relative error to* ∞. *If a quartile is larger than* 100, *we remove it from the plot. There are few differences among those algorithms on* A_1, A_3, *and* A_4. *UNIF and NOCD are clearly inferior to algorithms that explore both conditioning and leverage score-based sampling. UNIF and NOCD also failed on* A_2 *completely. GT and FJLT failed on* A_2 *when the sample size is smaller than* 512. *CT works slightly worse than FCT*1 *and FCT*2 *on these tests. One interesting fact from the result is that we see* $\varepsilon \sim 1/s$ *instead of* $1/s^{1/2}$.

with FCT1 and FCT2 suggests the use of these transforms in larger-scale applications of ℓ_1 regression.

6.3. Evaluation on a large-scale ℓ_1 regression problem. Here, we continue to demonstrate the capability of sampling-based algorithms in large-scale applications by solving a large-scale ℓ_1 regression problem with imbalanced and corrupted measurements. The problem is of size $5.24e9 \times 15$, generated in the following way:

1. The true signal x^* is a standard Gaussian vector.
2. Each row of the design matrix A is a canonical vector, which means that we only estimate a single entry of x^* in each measurement. The number of measurements on the ith entry of x^* is twice as large as that on the $(i+1)$st entry, $i = 1, \ldots, 14$. We have 2.62 billion measurements on the first entry while only 0.16 million measurements on the last. Imbalanced measurements apparently create difficulties for sampling-based algorithms.

TABLE 4

The first and the third quartiles of relative errors in 1-, 2-, and ∞ norms. CT clearly performs the best. GT follows closely. NOCD generates large errors, while UNIF works but it is about a magnitude worse than CT.

	$\frac{\|x-x^*\|_1}{\|x^*\|_1}$	$\frac{\|x-x^*\|_2}{\|x^*\|_2}$	$\frac{\|x-x^*\|_\infty}{\|x^*\|_\infty}$
CT	[0.008, 0.0115]	[0.00895, 0.0146]	[0.0113, 0.0211]
GT	[0.0126, 0.0168]	[0.0152, 0.0232]	[0.0184, 0.0366]
NOCD	[0.0823, 22.1]	[0.126, 70.8]	[0.193, 134]
UNIF	[0.0572, 0.0951]	[0.089, 0.166]	[0.129, 0.254]

3. The response vector is given by

$$b_i = \begin{cases} 1000\varepsilon_i & \text{with probability } 0.001, \\ a_i^T x^* + \varepsilon_i & \text{otherwise,} \end{cases} \qquad i = 1, \ldots,$$

where a_i is the ith row of A and $\{\varepsilon_i\}$ are i.i.d. samples drawn from the Laplace distribution. 0.1% measurements are corrupted to simulate noisy real-world data. Due to these corrupted measurements, ℓ_2 regression will not give us an accurate estimate, and ℓ_1 regression is certainly a more robust alternative.

Since the problem is separable, we know that an optimal solution is simply given by the median of responses corresponding to each entry.

The experiments were performed on a Hadoop cluster with 40 cores. Similar to our previous test, we implemented and compared Cauchy-conditioned sampling (CT), Gaussian-conditioned sampling (GT), unconditioned sampling (NOCD), and uniform sampling (UNIF). Since A only has $2n$ nonzeros, CT takes $O(nd \log d)$ time instead of $O(nd^2 \log d)$, which makes it the fastest among CT, FCT1, and FCT2 on this particular problem. Moreover, even if A is dense, data at this scale are usually stored on secondary storage, and thus time spent on scanning the data typically dominates the overall running time. Therefore, we only implemented CT for this test. Note that the purpose of this test is not to compare CT, FCT1, and FCT2 (which we did above), but to reveal some inherent differences among ℓ_1 conditioned sampling (CT, FCT1, and FCT2), ℓ_2 conditioned sampling (GT and FJLT), and other sampling algorithms (NOCD and UNIF). For each algorithm, we sample approximately 100000 (0.019%) rows and repeat the sampling 100 times, resulting in 100 approximate solutions. Note that those 100 approximate solutions can be computed simultaneously in a single pass.

We first check the overall performance of these sampling algorithms, measured by relative errors in 1-, 2-, and ∞ norms. The results are shown in Table 4. Since the algorithms are all randomized, we show the first and the third quartiles of the relative errors in 100 independent runs. We see that CT clearly performs the best, followed by GT. UNIF works, but it is about a magnitude worse than CT. NOCD is close to UNIF at the first quartile but makes very large errors at the third. Without conditioning, NOCD is more likely to sample outliers because the response from a corrupted measurement is much larger than that from a normal measurement. However, those corrupted measurements contain no information about x^*, which leads to NOCD's poor performance. UNIF treats all the measurements the same, but the measurements are imbalanced. Although we sample 100000 measurements, the expected number of measurements on the last entry is only 3.05, which downgrades UNIF's overall performance.

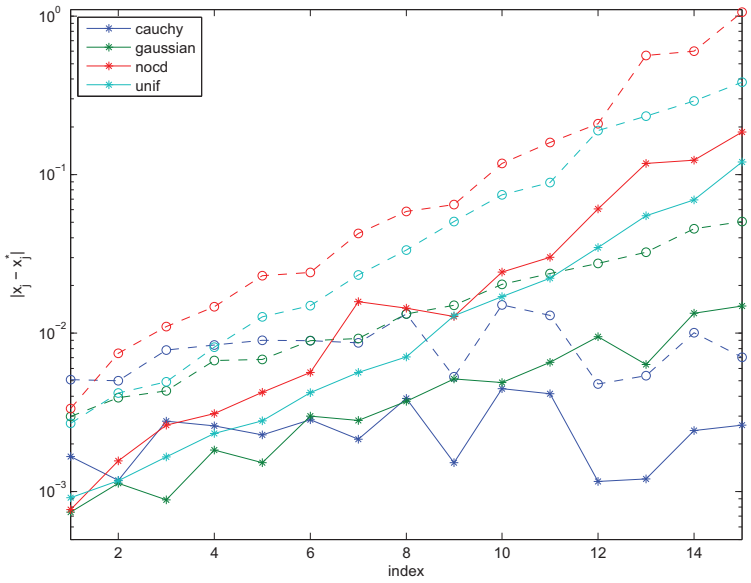

FIG. 6. *The first (solid) and the third (dashed) quartiles of entrywise absolute errors for our large-scale ℓ_1 regression empirical evaluation. See the text for details. Color is available in the online version and from* http://arxiv.org/abs/1207.4684.

We continue to analyze entrywise errors. Figure 6 draws the first and the third quartiles of entrywise absolute errors, which clearly reveals the differences among ℓ_1 conditioned sampling, ℓ_2 conditioned sampling, and other sampling algorithms. While UNIF samples uniformly rowwise, CT tends to sample uniformly entrywise. Although not as good as other algorithms on the first entry, CT maintains the same error level across all the entries, delivering the best overall performance. The ℓ_2-based GT sits between CT and UNIF. ℓ_2 conditioning can help detect imbalanced measurements to a certain extent and adjust the sampling weights accordingly, but it is still biased towards the measurements on the first several entries.

To summarize, we have shown that ℓ_1 conditioned sampling indeed works on large-scale ℓ_1 regression problems and its performance looks promising. We obtained about two accurate digits (0.01 relative error) on a problem of size $5.24e9 \times 15$ by passing over the data twice and sampling only 100000 (0.019%) rows in a judicious manner.

7. Conclusion. We have introduced the Fast Cauchy transform, an ℓ_1-based analogue of Fast Hadamard–based random projections. We have also demonstrated that this fast ℓ_1-based random projection can be used to develop algorithms with improved running times for a range of ℓ_1-based problems; we have provided extensions of these results to ℓ_p; and we have provided the first implementation and empirical evaluation of an ℓ_1-based random projection. Our empirical evaluation clearly demonstrates that for large and very rectangular problems, for which low-precision solutions are acceptable, our implementation follows our theory quite well; and it also points to interesting connections between ℓ_1-based projections and ℓ_2-based projections in practical settings. Understanding these connections theoretically, exploiting other properties such as sparsity, and using these ideas to develop improved algorithms for high-precision solutions to large-scale ℓ_1-based problems are important future direc-

tions raised by our work.

Appendix A. Proofs of technical Cauchy lemmas.

A.1. Proof of Lemma 3 (Cauchy upper tail inequality). The proof uses techniques similar to the bounds due to Indyk [15] for sums of independent clipped half-Cauchy random variables. Fix $M > 0$ (we will choose M later), and define the events

$$F_i = \{|C_i| \leq M\},$$

and $F = \cap_{i\in[m]} F_i$. Note that $F \cap F_i = F$. Using the pdf of a Cauchy and because $\tan^{-1} x \leq x$, we have that

$$\mathbf{Pr}[F_i] = \frac{2}{\pi}\tan^{-1}(M) = 1 - \frac{2}{\pi}\tan^{-1}\left(\frac{1}{M}\right) \geq 1 - \frac{2}{\pi M}.$$

By a union bound, $\mathbf{Pr}[F] \geq 1 - \frac{2m}{\pi M}$. Further, $\mathbf{Pr}[F|F_i]\mathbf{Pr}[F_i] = \mathbf{Pr}[F \cap F_i] = \mathbf{Pr}[F]$, and hence $\mathbf{Pr}[F|F_i] = \mathbf{Pr}[F]/\mathbf{Pr}[F_i]$. We now bound $\mathbf{E}\left[|C_i| \;\middle|\; F\right]$. First, observe that

$$\begin{aligned}\mathbf{E}\left[|C_i| \;\middle|\; F_i\right] &= \mathbf{E}\left[|C_i| \;\middle|\; F_i \cap F\right]\mathbf{Pr}[F|F_i] + \mathbf{E}\left[|C_i| \;\middle|\; F_i \cap \bar{F}\right]\mathbf{Pr}[\bar{F}|F_i] \\ &\geq \mathbf{E}\left[|C_i| \;\middle|\; F_i \cap F\right]\mathbf{Pr}[F|F_i].\end{aligned}$$

Next, since $F_i \cap F = F$, we have that

$$\mathbf{E}\left[|C_i| \;\middle|F\right] \leq \frac{\mathbf{E}\left[|C_i| \;\middle|\; F_i\right]}{\mathbf{Pr}[F|F_i]} = \frac{\mathbf{E}\left[|C_i| \;\middle|\; F_i\right]\mathbf{Pr}[F_i]}{\mathbf{Pr}[F]}.$$

Finally, by using the pdf of a Cauchy, $\mathbf{E}\left[|C_i|\middle|F_i\right] = \frac{1}{\pi}\log(1+M^2)/\mathbf{Pr}[F_i]$, and so

$$\mathbf{E}\left[|C_i| \;\middle|F\right] \leq \frac{\frac{1}{\pi}\log(1+M^2)}{\mathbf{Pr}[F]} \leq \frac{\frac{1}{\pi}\log(1+M^2)}{1-2m/\pi M}.$$

We conclude that

$$\mathbf{E}[X|F] = \sum_{i\in[m]} \gamma_i \mathbf{E}\left[|C_i| \;\middle|F\right] \leq \frac{\gamma}{\pi}\cdot\frac{\log(1+M^2)}{1-2m/\pi M}.$$

By Markov's inequality and because $\mathbf{Pr}[X \geq \gamma t|\bar{F}] \leq 1$, we have

$$\begin{aligned}\mathbf{Pr}[X \geq \gamma t] &= \mathbf{Pr}[X \geq \gamma t|F]\mathbf{Pr}[F] + \mathbf{Pr}[X \geq \gamma t|\bar{F}](1-\mathbf{Pr}[F]) \\ &\leq \frac{1}{\pi t}\cdot\frac{\log(1+M^2)}{1-2m/\pi M} + \frac{2m}{\pi M}.\end{aligned}$$

The result follows by setting $M = 2mt$.

A.2. Proof of Lemma 4 (Cauchy lower tail inequality). To bound the lower tail, we will use Lemma 1. By homogeneity, it suffices to prove the result for $\gamma = 1$. Let $Z_i = \gamma_i \min(|C_i|, M)$. Clearly, $Z_i \leq \gamma_i|C_i|$, and so defining $Z = \sum_i Z_i$, we have that $Z \leq X$ and $\mathbf{Pr}[X \leq 1-t] \leq \mathbf{Pr}[Z \leq 1-t]$. Thus, we have that

$$\mathbf{Pr}[Z \leq 1-t] = \mathbf{Pr}[Z \leq \mathbf{E}[Z] - (\mathbf{E}[Z] - 1 + t)] \leq \exp\left(\frac{-(\mathbf{E}[Z]-1+t)^2}{2\sum_i \mathbf{E}[Z_i^2]}\right),$$

where the last step holds by Lemma 1 for $1 - t < \mathbf{E}[Z]$. Using the distribution of the half-Cauchy, one can verify using standard techniques that by choosing $M \approx 1.6768$, $\mathbf{E}[Z_i] = \gamma_i$ and $\mathbf{E}[Z_i^2] \leq \frac{3}{2}\gamma_i^2$, so $\sum_i \mathbf{E}[Z_i] = 1$ and $\sum_i \mathbf{E}[Z_i^2] \leq \frac{3}{2}\sum_i \gamma_i^2 \leq \frac{3}{2\beta^2}$. It follows that $\mathbf{Pr}[Z \leq 1-t] \leq \exp\left(-t^2/\frac{3}{\beta^2}\right)$, and the result follows.

A.3. Proof of Lemma 5 (ℓ_1 sampling lemma). First, note that $\|DZx\|_1 = \sum_{i\in[n]} D_{ii}|Z_{(i)}x|$, and since $\mathbf{E}[D_{ii}] = 1$, $\mathbf{E}[\|DZx\|_1] = \sum_{i\in[n]} |Z_{(i)}x| = \|Zx\|_1$. Next, observe that

$$\sum_{i\in[n]} D_{ii}|Z_{(i)}x| - \sum_{i\in[n]} |Z_{(i)}x| = \sum_{\hat{p}_i<1} D_{ii}|Z_{(i)}x| - \sum_{\hat{p}_i<1} |Z_{(i)}x|$$

because when $\hat{p}_i = 1$, that row must be sampled and hence does not contribute to the deviation. So, we only need to analyze the RHS of the above equation. From now on, we only consider those i with $\hat{p}_i < 1$, in which case $\hat{p}_i = s{\cdot}t_i$, where $t_i \geq a\|Z_{(i)}\|_1/\|Z\|_1$. Let Q_i be the (positive) random variable $D_{ii}|Z_{(i)}x|$; either $Q_i = 0$ or

$$Q_i = \frac{|Z_{(i)}x|}{\hat{p}_i} \leq \frac{\|Z_{(i)}\|_1\|x\|_\infty}{\hat{p}_i} = \frac{\|Z_{(i)}\|_1\|x\|_\infty}{st_i} \leq \frac{1}{as}\|Z\|_1\|x\|_\infty = \frac{\gamma}{s},$$

where we defined $\gamma = \frac{1}{a}\|Z\|_1\|x\|_\infty$. We can also obtain a bound for $\sum_{\hat{p}_i<1} \mathbf{Var}[Q_i]$:

$$\sum_{\hat{p}_i<1} \mathbf{Var}[Q_i] = \sum_{\hat{p}_i<1} \mathbf{Var}[Q_i] \leq \sum_{\hat{p}_i<1} \mathbf{E}[Q_i^2] = \sum_{\hat{p}_i<1} \frac{|Z_{(i)}x|^2}{\hat{p}_i} = \sum_{\hat{p}_i<1} Q_i|Z_{(i)}x| \leq \frac{\gamma}{s}\|Zx\|_1,$$

where, in the last inequality, we used the upper bound for Q_i and we further upper bounded by summing over all $i \in [n]$. Let $Q = \sum_i Q_i$ with $Q_i \leq \gamma$; the standard Bernstein bound states that

$$\mathbf{Pr}\left[|Q - \mathbf{E}[Q]| > \varepsilon\right] \leq 2\exp\left(\frac{-\varepsilon^2}{2\sum_i \mathbf{Var}[Q_i] + \frac{2}{3}\varepsilon\gamma}\right).$$

Plugging in our bounds for $\sum_i \mathbf{Var}[Q_i]$ and γ, we deduce that

$$\mathbf{Pr}\Big[\big|\|DZx\|_1 - \|Zx\|_1\big| > \varepsilon\|Zx\|_1\Big] \leq 2\exp\left(\frac{-\varepsilon^2\|Zx\|_1^2}{\frac{2\gamma}{s}\|Zx\|_1 + \frac{2\varepsilon\gamma}{3s}\|Zx\|_1}\right).$$

The lemma follows after some simple algebraic manipulations.

Appendix B. Proof of Theorem 4 (fast ℓ_1 well-conditioned basis). Clearly, $U = AR^{-1}$ is in the range of A and has the same null-space; otherwise $\Pi_1 A$ would not preserve lengths to relative error. Therefore U is a basis for the range of A. Consider any $x \in \mathbb{R}^d$. The first claim of the theorem follows from the following derivations:

$$\|U\|_1 = \|AR^{-1}\|_1 \overset{(a)}{\leq} \|\Pi_1 AR^{-1}\|_1 \leq \sqrt{r_1}\|\Pi_1 AR^{-1}\|_2 \overset{(b)}{=} d\sqrt{r_1};$$

$$\|x\|_\infty \leq \|x\|_2 \overset{(b)}{=} \|\Pi_1 AR^{-1}x\|_2 \leq \|\Pi_1 AR^{-1}x\|_1 \overset{(c)}{\leq} \kappa\|AR^{-1}x\|_1 = \kappa\|Ux\|_1.$$

(a) follows from the lower bound in (7) because it holds for every column of AR^{-1}; (b) follows because by the construction of R, $\Pi_1 AR^{-1}$ has d orthonormal columns; finally, (c) follows from the upper bound in (7).

Finally, to obtain the corollary, if Π_1 satisfying (7) is constructed using Theorem 2 with small fixed probability of failure δ, then $r_1 = O(d\log d)$ and $\kappa = O(d^{2+\eta}\log d)$. The running time to compute R^{-1} is obtained by summing $O(nd\log d)$ (to compute $\Pi_1 A$) and $O(r_1 d^2) = O(d^3\log d)$ (to obtain $R^{-1} \in \mathbb{R}^{d\times d}$).

Appendix C. Proof of Theorem 5 (Fast Cauchy ℓ_1 regression). For $X \in \mathbb{R}^{n\times q}$, we analyze the more general constrained ℓ_1 regression problem, $\min_{x\in\mathcal{C}} \|Xx\|_1$,

where $\mathcal{C} \subseteq \mathbb{R}^q$ is a convex set, and we show that $\hat{x} \in \mathcal{C}$ constructed by our algorithm is a $(1+\varepsilon)$-approximation for this more general problem:

$$\|X\hat{x}\|_1 \leq (1+\varepsilon) \min_{x \in \mathcal{C}} \|Xx^*\|_1.$$

(That is, we actually prove a somewhat stronger result than we state in Theorem 5. This more general problem involves calling our main algorithm with $b = \{\}$ (NULL) and then incorporating the constraint that $x \in \mathcal{C}$ into the optimization problem that is solved as a black box in the last step. Of course, if the constraint set is not a polytope, then the last step may involve more sophisticated techniques than linear programming.) The classic ℓ_1 regression is obtained by setting $X = \begin{bmatrix} A & -b \end{bmatrix}$ $(q = d+1)$ with constraint $\mathcal{C} = \{x : e_{d+1}^T x = 1\}$, which corresponds to setting $x_{d+1} = 1$.

The main ingredients in our proof follow along a line similar to those in [6, 7, 24]. We use the notation from Figure 2. From step 1, Π_1 satisfies (7) with $A \leftarrow X$; i.e., Π_1 preserves the ℓ_1 norm of vectors in the range of X:

$$\|Xx\|_1 \leq \|\Pi_1 Xx\|_1 \leq \kappa \|Xx\|_1. \tag{9}$$

Let $\mathcal{C}' = \{y = R^{-1}x : x \in \mathcal{C}\}$ be a linear transform of the constraint set. We start with a basic lemma that allow us to use U instead of X. This lemma says that if we can construct a sampling matrix D for U under the constraint $\mathcal{C}'$ such that solving the downsampled problem for U gives a $(1+\varepsilon)$-approximation, then that *same* sampling matrix works for X under the constraint $\mathcal{C}$.

LEMMA 11. *Let $U = XR^{-1}$, and let D be any diagonal sampling matrix as in Figure 2. Suppose that for any $\hat{y} \in \mathcal{C}'$ that minimizes $\|DUy\|$, $\hat{y}$ is a $(1+\varepsilon)$-approximation for the problem $\min_{y \in \mathcal{C}'} \|Uy\|$. Let $\hat{x}$ be any solution to $\min_{x \in \mathcal{C}} \|DXx\|$. Then, $\hat{x}$ is a $(1+\varepsilon)$-approximation for the problem $\min_{x \in \mathcal{C}} \|Xx\|$.*

Proof. Select $\hat{y} = R^{-1}\hat{x} \in \mathcal{C}'$. For any $y \in \mathcal{C}'$, there is some $x \in \mathcal{C}$ with $y = R^{-1}x$, and we have

$$\|DUy\|_1 = \|DUR^{-1}x\|_1 = \|DXx\|_1 \overset{(a)}{\geq} \|DX\hat{x}\|_1 = \|DUR^{-1}\hat{x}\|_1 = \|DU\hat{y}\|_1,$$

where (a) is by the optimality of $\hat{x}$. So, $\hat{y}$ minimizes $\|DUy\|_1$, and hence for all $y \in \mathcal{C}'$, $\|U\hat{y}\|_1 \leq (1+\varepsilon)\|Uy\|_1$. Now consider any $x \in \mathcal{C}$ and let $y = R^{-1}x \in \mathcal{C}'$. Then,

$$\|X\hat{x}\|_1 = \|UR^{-1}\hat{x}\|_1 = \|U\hat{y}\|_1 \leq (1+\varepsilon)\|Uy\|_1 = (1+\varepsilon)\|UR^{-1}x\|_1 = (1+\varepsilon)\|Xx\|_1,$$

completing the proof. ∎

Remarks. We emphasize that our proof accommodates an *arbitrary* constraint set $\mathcal{C}$. For the classical ℓ_1 regression problem, that is of interest to us in Theorem 5, we only need $\mathcal{C}$ to be specified by a single linear constraint. In what follows, we will work with U and show that our algorithm generates a coreset that works, regardless of the constraint set $\mathcal{C}'$.

By Theorem 4, $U = XR^{-1}$ is an (α, β)-conditioned basis for the range of X, where

$$\alpha \leq q\sqrt{r_1} \quad \text{and} \quad \beta \leq \kappa.$$

So, $\|U\|_1 \leq q\sqrt{r_1}$ and for all $y \in \mathbb{R}^q$, $\|y\|_\infty \leq \kappa \|Uy\|_1$. We next show that λ_i estimates $\|U_{(i)}\|_1$. The following lemma is a straightforward application of a Chernoff bound to independent half-Cauchys (see also Claims 1 and 2 and Lemmas 1 and 2 in [15]).

LEMMA 12. *Let $Z_1, \ldots, Z_{r_2}$ be r_2 independent Cauchys. Then,*

$$\frac{1}{2} \leq \text{median}\{|Z_1|, \ldots, |Z_{r_2}|\} \leq \frac{3}{2}$$

with probability at least $1 - 2e^{-cr_2}$, where $c \geq 2(\tan^{-1}(\frac{1}{5}))^2 \geq 0.07$ is a constant.

Fix i, and for $j \in [r_2]$ define the random variables $Z_j = \Lambda_{ij}$ to apply Lemma 12. Observe that for $j \in [r_2]$, $Z_j = \Lambda_{ij} = U_{(i)}\Pi_2^{(j)}$ are independent Cauchy random variables scaled by $\|U_{(i)}\|_1$. Applying Lemma 12 with $\lambda_i = \text{median}_{j \in r_2} |\Lambda_{ij}|$, we have that with probability at least $1 - 2e^{-cr_2}$,

$$\tfrac{1}{2}\|U_{(i)}\|_1 \leq \lambda_i \leq \tfrac{3}{2}\|U_{(i)}\|_1. \tag{10}$$

By a union bound, these inequalities hold for all $i \in [n]$ with probability at least $1 - 2ne^{-cr_2} \geq 1 - \delta$ for $r_2 \geq \frac{1}{c} \log \frac{2n}{\delta}$ (since $\frac{1}{c} \leq 15$, our algorithm satisfies this condition). Next we show that if the sampling matrix preserves the ℓ_1 norm of every vector in the range of U, then we are done.

LEMMA 13. *Given D with n columns, suppose that for all $y \in \mathbb{R}^q$,*

$$(1 - \varepsilon)\|Uy\|_1 \leq \|DUy\|_1 \leq (1 + \varepsilon)\|Uy\|_1, \tag{11}$$

and suppose that $\hat{y}$ is a solution to $\min_{y \in \mathcal{C}'} \|DUy\|$. Then, for all $y \in \mathcal{C}'$,

$$\|U\hat{y}\|_1 \leq \left(\frac{1 + \varepsilon}{1 - \varepsilon}\right) \|Uy\|_1.$$

Proof. Since D preserves norms, for any $y \in \mathcal{C}'$, we have that

$$\|U\hat{y}\|_1 \leq \frac{1}{1 - \varepsilon}\|DU\hat{y}\|_1 \overset{\text{(a)}}{\leq} \frac{1}{1 - \varepsilon}\|DUy\|_1 \leq \frac{1 + \varepsilon}{1 - \varepsilon}\|Uy\|_1.$$

(a) is by the optimality of $\hat{y}$. ∎

The remainder of the proof is to show that D from our algorithm in Figure 2 satisfies the precondition of Lemma 13, namely (11). We need two ingredients. The first is the ℓ_1-sampling lemma (Lemma 5). The second ingredient is a standard γ-net argument.

We are going to apply Lemma 5 with $Z = U$. From (10) (which holds for all $i \in [n]$ with probability at least $1 - \delta$), $\lambda_i / \sum_{i \in [n]} \lambda_i \geq \frac{1}{3}\|U_{(i)}\|_1 / \|U\|_1$, and so we can apply Lemma 5 with $a = \frac{1}{3}$. Since U is (α, β)-conditioned, $\|Uy\|_1 \geq \frac{1}{\beta}\|y\|_\infty$, and $\|U\|_1 \leq \alpha$, and so we have that with probability at least $1 - \delta$,

$$(1 - \varepsilon)\|Uy\|_1 \leq \|DUy\|_1 \leq (1 + \varepsilon)\|Uy\|_1,$$

where $\delta \leq 2\exp\left(\frac{-s\varepsilon^2}{(6+2\varepsilon)\alpha\beta}\right)$, and $\alpha\beta \leq \kappa q\sqrt{r_1}$. If $y = \mathbf{0}$, then the bounds trivially hold; by rescaling, it thus suffices to show the bound for all $y \in \mathbb{R}^q$ with $\|y\|_\infty = 1$. We now show this using a standard γ-net argument. Consider the uniform lattice on $\mathbb{R}^q$ specified by $T = \frac{\gamma}{q}\mathbb{Z}^q$ (we assume that q/γ is a positive integer for simplicity). Let $H = \{z : \|z\|_\infty \leq 1\} \cap T$ be the restriction of this grid to only its points within the hypercube of points with ℓ_∞ norm equal to 1; $|H| \leq (\frac{2q}{\gamma})^q$. Consider any y with $\|y\|_\infty = 1$, and let h be the closest grid point in H to y. Then

$$y = h + \frac{\gamma}{q}\sum_{i=1}^{q} \zeta_i e_i, \tag{12}$$

where $0 \le |\zeta_i| \le 1$. Observe that $e_i \in H$. By a union bound, for every $h \in H$, with probability at least $1-\delta$,

$$(1-\varepsilon)\|Uh\|_1 \le \|DUh\|_1 \le (1+\varepsilon)\|Uh\|_1,$$

where $\delta \le 2|H| \exp\left(\frac{-s\varepsilon^2}{(6+2\varepsilon)\alpha\beta}\right)$. We condition on this high probability event. Then,

$$\begin{aligned}\|DUy\|_1 = \left\|DUh + \frac{\gamma}{q}\sum_{i=1}^{q}\zeta_i DUe_i\right\|_1 &\le \|DUh\|_1 + \frac{\gamma}{q}\sum_{i=1}^{q}|\zeta_i|\|DUe_i\|_1 \\ &\le (1+\varepsilon)\left(\|Uh\|_1 + \frac{\gamma}{q}\sum_{i=1}^{q}\|Ue_i\|_1\right).\end{aligned}$$

Applying U to both sides of (12) and using the triangle inequality, $\|Uh\|_1 \le \|Uy\|_1 + \frac{\gamma}{q}\sum_{i=1}^{q}|\zeta_i|\|Ue_i\|_1$. We conclude that

$$\|DUy\|_1 \le (1+\varepsilon)\left(\|Uy\|_1 + \frac{2\gamma}{q}\sum_{i=1}^{q}\|Ue_i\|_1\right) \le (1+\varepsilon)\|Uy\|_1\left(1+\frac{2\gamma\alpha\beta}{q}\right),$$

where we used $\|Uy\|_1 \ge \frac{1}{\beta}\|y\|_\infty = \frac{1}{\beta}$ (since $\|y\|_\infty = 1$) and $\sum_{i=1}^{q}\|Ue_i\|_1 = \|U\|_1 \le \alpha$. In an analogous way, we get the lower bound:

$$\begin{aligned}\|DUy\|_1 = \left\|DUh + \frac{\gamma}{q}\sum_{i=1}^{q}\zeta_i DUe_i\right\|_1 &\ge \|DUh\|_1 - \frac{\gamma}{q}\sum_{i=1}^{q}|\zeta_i|\|DUe_i\|_1 \\ &\ge (1-\varepsilon)\|Uh\|_1 - \frac{\gamma(1+\epsilon)}{q}\sum_{i=1}^{q}\|Ue_i\|_1 \\ &= (1-\varepsilon)\|Uh\|_1 - \frac{\gamma(1+\epsilon)}{q}\|U\|_1.\end{aligned}$$

Again, applying U to (12) and using the triangle inequality gives $\|Uh\|_1 \ge \|Uy\|_1 - \frac{\gamma}{q}\|U\|_1$. Further, $\|Uy\|_1 \le \|U\|_1\|y\|_\infty \le \alpha$, and so we have

$$\|DUy\|_1 \ge (1-\varepsilon)\left(\|Uy\|_1 - \frac{2\gamma}{q(1-\epsilon)}\|U\|_1\right) \ge (1-\varepsilon)\|Uy\|_1\left(1-\frac{4\gamma\alpha\beta}{q}\right),$$

where we assume $\epsilon \le \frac{1}{2}$. Setting $\gamma = q\varepsilon/(4\alpha\beta)$, using $(1+\varepsilon)^2 \le 1+3\varepsilon$ and $(1-\varepsilon)^2 \ge 1-3\varepsilon$ (for $\varepsilon < \frac{1}{2}$), and rescaling ε by dividing by 3, we obtain that with probability at least $1-\delta$,

$$(1-\varepsilon)\|Uy\|_1 \le \|DUy\|_1 \le (1+\varepsilon)\|Uy\|_1,$$

where $\delta \le 2|H| \exp\left(\frac{-s\varepsilon^2}{9(6+2\varepsilon/3)\alpha\beta}\right)$, and $|H| \le (\frac{24\alpha\beta}{\varepsilon})^q$. Solving for s using $\alpha \le q\sqrt{r_1}$ and $\beta \le \kappa$, and simplifying a little, we require

$$s \ge \frac{63\kappa q\sqrt{r_1}}{\varepsilon^2}\left(q\log\frac{24\kappa q\sqrt{r_1}}{\varepsilon} + \log\frac{2}{\delta}\right).$$

The total success probability is $1-2\delta$, which results from a union bound applied to the two random processes involving Π_2 and D. The theorem now follows by setting $\delta = 1/d^\rho$ for a constant ρ. This concludes the proof of the correctness.

Running time. Set $\delta = \frac{1}{3q^\rho}$ for $\rho = O(1)$. We compute the running time as follows. In step 2, if we use Theorem 2 for Π_1 (which succeeds with probability $1-\delta$), the time to compute $\Pi_1 X$ is $O(nq \log q)$ and $r_1 = O(q \log q)$ and $\kappa = O\left(q^{\rho+2} \log q)\right)$ (r_1 and κ affect the running time of later steps). We need to compute an orthogonal factorization in $O(r_1 q^2)$ and then compute R^{-1} in $O(q^3)$ for a total run time of step 2 that is $O((n+q^2)q \log q)$. In step 3, $r_2 = O(\log n)$ by our choice of r_2, so the time to compute $\Lambda = XR^{-1}\Pi_2$ is in $O(nqr_2 + r_2 q^2) = O(nq \log n + q^2 \log n)$, where $O(nq \log n + q^2 \log n)$ is the time needed to compute $R^{-1}\Pi_2$ followed by $X \cdot (R^{-1}\Pi_2)$. Note that $q^2 \log n \le nq \log n$.

Since computation of the median of r_2 elements is in $O(r_2)$, computing all λ_i takes $O(nr_2) = O(n \log n)$ time. Thus, the running time for steps 1–5 is $O(nq \log n) + q^3 \log q$.

In step 6, $s = O\big(\varepsilon^{-2} q^{\rho+\frac{9}{2}} \log^{\frac{5}{2}}(\frac{q}{\varepsilon})\big)$. It takes $O(n)$ time to sample the diagonal matrix D and then $O(qS)$ time to construct DA and Db, where S is the number of nonzero entries in D. Finally, step 8 takes $\phi(S,d) = \Omega(dS)$ time to solve the ℓ_1 regression on the smaller problem with s rows in d dimensions. The total running time is thus

$$O\left(nq \log n + \phi(S,q)\right),$$

where $\mathbf{E}[S] \le s = O\big(\varepsilon^{-2} q^{\rho+\frac{9}{2}} \log^{\frac{5}{2}}(\frac{q}{\varepsilon})\big)$ and S is very tightly concentrated around s (via a standard Bernstein bound) because it is the sum of independent binomial variables; specifically, with probability at least $1 - e^{-\frac{3}{8}s}$, $S \le 2s$. Hence $S = O(s)$ with probability $1 - o(1)$. The probability of success is $1 - 3\delta = 1 - \frac{1}{q^\rho}$ (union bound). Since $s = \varepsilon^{-2} q^{\rho+\frac{9}{2}} \operatorname{poly}(\log \frac{q}{\varepsilon})$, and since standard algorithms for linear programming give $\phi(S,q) = Sq^{O(1)}$, we have the result claimed in the theorem for $q = O(d)$.

Appendix D. Proof of Theorem 6 (optimized Fast Cauchy ℓ_1 regression). As in the proof of Theorem 5 in Appendix C, given are $X \in \mathbb{R}^{n \times q}$ and the constraint set $\mathcal{C}$. We condition on $\Pi_1 \in \mathbb{R}^{r_1 \times n}$ satisfying (7) and Theorem 4. So, $U = XR^{-1}$ is $(q\sqrt{r_1}, \kappa)$-conditioned where r_1 and κ depend on n, q, δ (this holds with probability at least $1-\delta$). Thus, $\|U\|_1 \le q\sqrt{r_1}$ and $\|x\|_\infty \le \kappa \|Ux\|_1$ for any $x \in \mathbb{R}^q$. It follows that

$$\frac{q}{\kappa} \le \|U\|_1 \le q\sqrt{r_1}.$$

(The lower bound follows from $\sum_{i\in[q]} \|e_i\|_\infty \le \sum_{i\in[q]} \|Ue_i\|_1$, where e_i are standard basis vectors.) In the proof of Theorem 5, we proved the following result. Given *weights* t_i, with

$$t_i \ge a \cdot \frac{\|U_{(i)}\|_1}{\|U\|_1} \qquad \forall i \in [n], \tag{13}$$

define leverage probabilities

$$\ell_i = \min\left(1, s \cdot t_i\right),$$

with

$$s \ge \frac{63\kappa q\sqrt{r_1}}{a\varepsilon^2} \cdot \left(q \log \frac{4q\sqrt{r_1}\max(q\sqrt{r_1}, \kappa)}{\varepsilon} + \log \frac{2}{\delta}\right), \tag{14}$$

and construct the random diagonal sampling matrix D with $D_{ii} = 1/\ell_i$ with probability ℓ_i and zero otherwise. Then, with probability at least $1-\delta$, solving the coreset problem given by DX and the constraints $\mathcal{C}$ gives a $(1+\varepsilon)/(1-\varepsilon)$ approximate solution

to the full L_1 regression with X and $\mathcal{C}$. In the proof of Theorem 5, the purpose of Π_2 was to allow us to construct weights t_i quickly such that with high probability, $\frac{1}{a} \leq 3$. Here, we show that our faster way to get weights results in only a $\text{poly}(d\epsilon^{-1}\log n)$ factor increase in $\frac{1}{a}$.

Recall $\Lambda = U\Pi_2$, where $\Pi_2 \in \mathbb{R}^{n\times r_2}$ is a matrix of i.i.d. standard Gaussian random variables, and $\hat{\lambda}_i = \text{median}_{j\in[r_2]} |\Lambda_{ij}|$, with

$$\hat{p}_i = \min\left(1, s\cdot\hat{\lambda}_i\right).$$

For $j \in [r_2]$, the Λ_{ij} are i.i.d. zero mean Gaussians with variance $\|U_{(i)}\|_2^2$, so $|\Lambda_{ij}|$ are i.i.d half-Gaussians. We need a result from [15].

LEMMA 14 (Lemma 2 of [15]). *Let $x_1, \ldots, x_r$ be i.i.d. with continuous distribution function F and $\lambda = \text{median}_{i\in[r]} x_i$. Then,*

$$\mathbf{Pr}\left[\lambda \geq F^{-1}(\tfrac{1}{2}-\epsilon)\right] \geq 1-\exp(-2\epsilon^2 r),$$
$$\mathbf{Pr}\left[\lambda \leq F^{-1}(\tfrac{1}{2}+\epsilon)\right] \geq 1-\exp(-2\epsilon^2 r).$$

For the half-Gaussian with variance σ^2, $F(x) = 2\phi(x/\sigma) - 1$, where ϕ is the standard Gaussian distribution function. Choosing $\epsilon = \frac{1}{4}$ and using Lemma 14, with probability at least $1-\exp(-r_2/2)$,

$$\hat{\lambda}_i \geq 0.3\cdot\|U_{(i)}\|_2,$$

where we used $0.3 < \phi^{-1}(\frac{5}{8})$. Using $\|U_{(i)}\|_2 \geq \|U_{(i)}\|_1/\sqrt{q}$ and $\|U\|_1 \geq q/\kappa$, it follows that

$$\hat{\lambda}_i \geq \frac{0.3}{\sqrt{q}}\|U_{(i)}\|_1 \geq \frac{0.3\sqrt{q}}{\kappa}\frac{\|U_{(i)}\|_1}{\|U\|_1} \tag{15}$$

holds with probability at least $1-2\exp(-r_2/2)$ for any particular i. If we required these bounds to hold for all $i \in [n]$, then to apply the union bound successfully, we would need to set $r_2 = \Omega(\log n)$, which is too costly. We want $r_2 = O(\log(d\epsilon^{-1}\log n))$, so we need a more subtle argument. We choose s as in (14) with $a = 0.3\sqrt{q}/\kappa$.

Let $t_i = \|U_{(i)}\|_1$ and $p_i = \min(1, s\cdot t_i)$ be sampling probabilities obtained from the exact L_1 leverage scores for U. For these sampling probabilities, a in (13) is larger, which would imply that a smaller s is needed. Nevertheless, any larger value of s will also work, and so the same value of s with $a = 0.3\sqrt{q}/\kappa$ will work with the weights t_i. Note that since Π_1 is fixed, p_i is not a random variable, but $\hat{p}_i$ is a random variable depending on Π_2. Fix r_2, and generate Π_2 and hence $\hat{\lambda}_i, \hat{p}_i$.

We define a set of indices $T \subseteq [n]$ as those i for which $\hat{\lambda}_i \geq (0.3\sqrt{q}/\kappa)\|U_{(i)}\|_1/\|U\|_1$. These are the indices for which Π_2 "worked." Essentially, these are the large leverage scores. The intuition behind our argument is that even though there may be some indices for which Π_2 did not work, there are enough large leverage scores for which Π_2 did work that the probability of these faulty indices coming into play is minuscule.

To make this argument, we define hybrid weights w_i to equal $\hat{\lambda}_i$ for $i \in T$ and t_i for $i \notin T$. By construction,

$$w_i \geq \frac{0.3\sqrt{q}}{\kappa}\frac{\|U_{(i)}\|_1}{\|U\|_1},$$

and so the same s works for constructing sampling probabilities $q_i = \min(1, s\cdot w_i)$. Note that in the algorithm, we do not actually construct (or know) p_i, q_i; they are just

used here as a hypothetical set of sampling probabilities which help us to analyze the performance of the actual sampling probabilities we use, which are $\hat{p}_i$. The important property about the q_i is that for $i \in T$, $q_i = \hat{p}_i$.

We call a set of rows that are sampled and rescaled according to a set of probabilities a *good* coreset if the coreset solution from this sample is a $(1+\varepsilon)$-approximation to the full L_1 regression. The sampling probabilities q_i give a good coreset with probability at least $1-\delta$. We now define several events over three random processes: Π_2, sampling a coreset according to $\hat{p}_i$, and sampling a coreset according to q_i. The last two random processes depend on the outcome of Π_2.

- AllBounded is the event $\{\hat{\lambda}_i \le C\sqrt{\log n} \cdot \|U_{(i)}\|_2 \ \forall i \in [n]\}$ (we will choose C later). We show that

$$\Pr_{\Pi_2}[\mathsf{AllBounded}] \ge 1 - \frac{1}{n^{\frac{1}{2}C^2-1}}.$$

  Indeed, $\hat{\lambda}_i$ is the median of r_2 i.i.d. zero mean Gaussians $x_1, \ldots, x_{r_2}$ with variance $\|U_{(i)}\|_2^2$, where

$$\mathbf{Pr}\left[x_i > C\sqrt{\log n}\|U_{(i)}\|_2\right] = 1 - \phi(C\sqrt{\log n}) \le 1/(\sqrt{\pi}n^{C^2/2})$$

  (by the properties of the Gaussian distribution). Define $z_i = 1$ if $x_i > C\sqrt{\log n}$ and 0 otherwise. Then $\hat{\lambda}_i > C\sqrt{\log n}$ if and only if $\sum_{i\in[r_2]} z_i > r_2/2$. We have $\mathbf{E}\left[\sum_{i\in[r_2]} z_i\right] \le r_2/(\sqrt{\pi}n^{C^2/2})$, and the result follows by a Markov bound and a union bound over $i \in [n]$.
- Let Good(q) be the event that the coresets sampled according to probabilities q_i are good.
- Let Good$(\hat{p})$ be the event that the coresets sampled according to $\hat{p}_i$ are good.
- Let BadRow be the event that either of the two coresets above contains a row $i \notin T$.

In what follows, we consider probabilities with respect to the joint distribution of Π_2 and the randomness of choosing the coresets according to q_i and $\hat{p}_i$.

$$\begin{aligned}\mathbf{Pr}[\mathsf{Good}(q)] &\le \mathbf{Pr}[\mathsf{BadRow}] + \mathbf{Pr}[\mathsf{Good}(q)|\neg\mathsf{BadRow}](1 - \mathbf{Pr}[\mathsf{BadRow}])\\ &= \mathbf{Pr}[\mathsf{BadRow}] + \mathbf{Pr}[\mathsf{Good}(\hat{p})|\neg\mathsf{BadRow}](1 - \mathbf{Pr}[\mathsf{BadRow}])\\ &\le \mathbf{Pr}[\mathsf{BadRow}] + \mathbf{Pr}[\mathsf{Good}(\hat{p})],\end{aligned}$$

where the second step follows because conditioning on ¬BadRow, the sampling probabilities $\hat{p}_i$ and q_i are identical (by construction). Thus,

$$\begin{aligned}\mathbf{Pr}[\mathsf{Good}(\hat{p})] &\ge \mathbf{Pr}[\mathsf{Good}(q)] - \mathbf{Pr}[\mathsf{BadRow}]\\ &\ge 1 - \delta - \mathbf{Pr}[\mathsf{BadRow}]\end{aligned}$$

because we know that the sampling probabilities q_i satisfy the conditions to get a good coreset with probability at least $1-\delta$. To get an upper bound on $\mathbf{Pr}[\mathsf{BadRow}]$, observe that

$$\begin{aligned}\mathbf{Pr}[\mathsf{BadRow}] &\le \Pr[\neg\mathsf{AllBounded}] + \Pr[\mathsf{BadRow}|\mathsf{AllBounded}]\\ &\le \frac{1}{n^{\frac{1}{2}C^2-1}} + \Pr[\mathsf{BadRow}|\mathsf{AllBounded}].\end{aligned}$$

To conclude, we obtain a bound on $\Pr[\mathsf{BadRow}|\mathsf{AllBounded}]$. Condition on Π_2 and that $\mathsf{AllBounded}$ holds. This fixes T and $\hat{p}_i$ and also means that $\hat{p}_i \leq C\sqrt{\log n}q_i$. Hence,

(16)
$$\Pr[\mathsf{BadRow}|\mathsf{AllBounded}] \leq \sum_{i\notin T}(\hat{p}_i+q_i) = (1+C\sqrt{\log n})\cdot\sum_{i\notin T} q_i = (1+C\sqrt{\log n})\cdot\sum_{i\notin T} p_i,$$

where the last equality is because $q_i = p_i$ for $i \notin T$. So the bound is determined by the sum of the leverage scores over the indices for which Π_2 did not work. This is the quantification of our intuition that the algorithm will work as long as Π_2 preserves enough of the large leverage scores. We need to bound $\sum_{i\notin T} p_i$, where T is a random set of indices depending on Π_2. We will use a Markov bound to bound $\sum_{i\notin T} p_i$ with high probability. We have

$$\begin{aligned}\mathbf{E}_{\Pi_2}\left[\sum_{i\notin T} p_i\right] &\geq \mathbf{E}_{\Pi_2}\left[\sum_{i\notin T} p_i \;\middle|\; \mathsf{AllBounded}\right]\mathbf{Pr}[\mathsf{AllBounded}]\\ &\geq \mathbf{E}_{\Pi_2}\left[\sum_{i\notin T} p_i \;\middle|\; \mathsf{AllBounded}\right]\cdot\left(1-\frac{1}{n^{\frac{1}{2}C^2-1}}\right).\end{aligned}$$

Since $\mathbf{Pr}[i \notin T] \leq \exp(-r_2/2)$,

$$\mathbf{E}_{\Pi_2}\left[\sum_{i\notin T} p_i\right] = \sum_{i\in[n]} p_i\cdot\mathbf{Pr}[i\notin T] \leq e^{-r_2/2}\sum_{i\in[n]} p_i.$$

But, $\sum_{i\in[n]} p_i \leq s\sum_{i\in[n]} t_i = s\|U\|_1 \leq sq\sqrt{r_1}$, where the last step follows from the conditioning of U which is assumed. Putting all this together,

$$\mathbf{E}_{\Pi_2}\left[\sum_{i\notin T} p_i \;\middle|\; \mathsf{AllBounded}\right] \leq \frac{sq\sqrt{r_1}e^{-r_2/2}}{1-n^{1-\frac{1}{2}C^2}}. \leq 2sq\sqrt{r_1}e^{-r_2/2},$$

where the last expression follows by setting $C = 2$, in which case $1 - 1/n \geq \frac{1}{2}$. Now, recalling $\rho > 0$ is given as in the theorem statement, if we set

$$r_2 = 2\log\left(2sq\sqrt{r_1}\log^{2\rho+1/2} n\right) = O\left(\log(d\epsilon^{-1}\log n)\right),$$

then $\mathbf{E}_{\Pi_2}\left[\sum_{i\notin T} p_i \;|\; \mathsf{AllBounded}\right] \leq 1/\log^{2\rho+1/2} n$. Applying a Markov bound and conditioning on $\mathsf{AllBounded}$, with probability at most $1/\log^{\rho} n$, the bound $\sum_{i\notin T} p_i > 1/\log^{\rho+1/2} n$ holds. Condition on this bad event not happening, in which case, using (16),

$$\Pr[\mathsf{BadRow}|\mathsf{AllBounded}] \leq \frac{1+2\sqrt{\log n}}{\log^{\rho+3/2} n} = O(1/\log^{\rho} n),$$

where we used $C = 2$. Using a union bound over this bad event not happening, we finally have that

$$\mathbf{Pr}[\mathsf{BadRow}] \leq \frac{1}{n} + \frac{1}{\log^{\rho+1/2} n} + \frac{1+2\sqrt{\log n}}{\log^{\rho+3/2} n},$$

from which $\mathbf{Pr}[\mathsf{Good}(\hat{p})] \geq 1 - \delta - O(\log^{-\rho} n)$. This completes the proof.

Appendix E. Proof of the fast ellipsoidal rounding theorems.

E.1. Proof of Theorem 9 (fast ellipsoidal rounding). For completeness, we state the following lemma, which is from [25] and which we will use in the proof of this theorem.

LEMMA 15 (Todd [25]). *Given an ellipsoid* $\mathcal{E} = \{u \in \mathbb{R}^d \,|\, u^T E^{-1} u \le 1\}$ *where* $E \in \mathbb{R}^{d\times d}$ *is symmetric positive definite and* $\mathcal{K} = \{u \in \mathbb{R}^d \,|\, -\beta(g^T E g)^{1/2} \le g^T u \le \beta(g^T E g)^{1/2}\}$ *for some* $g \in \mathbb{R}^d$*, the minimum-volume ellipsoid that contains* $\mathcal{E} \cap \mathcal{K}$ *is given by*

$$\mathcal{E}_+ = \begin{cases} \mathcal{E} & \text{if } \beta \ge d^{-1/2}, \\ \{u \in \mathbb{R}^d \,|\, u^T E_+^{-1} u \le 1\} & \text{if } 0 < \beta < d^{-1/2}, \end{cases}$$

where

$$E_+ = \delta\left(E - \sigma\frac{(Eg)(Eg)^T}{g^T E g}\right),$$
$$\delta = \frac{d(1-\beta^2)}{d-1}, \quad \sigma = \frac{1-d\beta^2}{1-\beta^2}.$$

When $\beta < d^{-1/2}$*, we have*

$$\frac{|\mathcal{E}_+|}{|\mathcal{E}|} = d^{1/2}\left(\frac{d}{d-1}\right)^{(d-1)/2}\beta(1-\beta^2)^{(d-1)/2}.$$

Now we proceed with the main part of the proof. We construct a sequence of ellipsoids $\mathcal{E}_1, \mathcal{E}_2, \ldots$, all centered at the origin, such that $\mathcal{E}_k \supseteq \mathcal{C}$ and $|\mathcal{E}_k|/|\mathcal{E}_{k-1}| < e^{3/8}/2,\ k = 1, 2, \ldots$, and thus this sequence must terminate in

$$\log(L^{-d})/\log(e^{3/8}/2) < 3.15d\log L$$

steps. Suppose we have $\mathcal{E}_k \supseteq \mathcal{C}$ centered at the origin. Determine all the extreme points of $\mathcal{E}_k$ along its axes. Let these points be $\pm x_{k,i},\ i = 1, \ldots, d$, and then check whether $\frac{1}{2\sqrt{d}} x_{k,i} \in \mathcal{C}$ for $i = 1, \ldots, d$. If all these points are in $\mathcal{C}$, so is their convex hull, denoted by $\mathcal{H}$. Apparently, $\frac{1}{2\sqrt{d}}\mathcal{E}_k$ is the LJ ellipsoid of $\mathcal{H}$, and hence shrinking $\frac{1}{2\sqrt{d}}\mathcal{E}_k$ by a factor $\frac{1}{\sqrt{d}}$ makes it contained in $\mathcal{H} \subseteq \mathcal{C}$. We have $\frac{1}{2d}\mathcal{E}_k \subseteq \mathcal{C} \subseteq \mathcal{E}_k$. Now suppose that $\frac{1}{2\sqrt{d}} x_{k,i_k} \notin \mathcal{C}$ for some i_k and the separation oracle returns $\mathcal{K}_k = \{x \in \mathbb{R}^d \,|\, -1 \le g_k^T x \le 1\}$ such that $\mathcal{C} \subseteq \mathcal{K}_k$ but $\frac{1}{2\sqrt{d}} x_{k,i_k} \notin \mathcal{K}_k$. Let $\mathcal{E}_{k+1}$ be the LJ ellipsoid of $\mathcal{E}_k \cap \mathcal{K}_k \supseteq \mathcal{C}$, which must be centered at the origin. Lemma 15 gives analytic formulas of $\mathcal{E}_{k+1}$ and $|\mathcal{E}_{k+1}|/|\mathcal{E}_k|$. Adopting the notation from Lemma 15, let $\mathcal{E}_k = \{x \in \mathbb{R}^d \,|\, x^T E_k^{-1} x \le 1\}$, and we have

$$(g_k^T E_k g_k)^{1/2} = \left[g_k^T\left(\sum_{i=1}^d x_{k,i} x_{k,i}^T\right) g_k\right]^{1/2}$$
$$\ge |g_k^T x_{k,i_k}| > 2\sqrt{d}.$$

The last inequality comes from the fact that $\frac{1}{2\sqrt{d}} x_{k,i_k} \notin \mathcal{K}_k$. Therefore,

$$\beta = (g_k^T E_k g_k)^{-1/2} < \frac{1}{2\sqrt{d}}$$

and

$$\frac{|\mathcal{E}_{k+1}|}{|\mathcal{E}_k|} < \frac{1}{2}\left(1 + \frac{3}{4d-4}\right)^{(d-1)/2} < e^{3/8}/2.$$

Thus, our construction is valid. For each step, it takes at most d calls to the separation oracle. Therefore, we need at most $3.15d^2 \log L$ calls to find a $2d$-rounding of $\mathcal{C}$. Computing the extreme points of $\mathcal{E}_k$ requires an eigendecomposition, which takes $O(d^3)$ time. Hence the total cost to find a $2d$-rounding is $3.15d^2 \log L$ calls and additional $O(d^4 \log L)$ time. We note that rank-one updates can be used for computing the eigendecomposition of $\mathcal{E}_k$ for efficiency. See Gu and Eisenstat [14].

E.2. Proof of Theorem 10. This is a direct consequence of Theorem 9. We present the proof for the case $p < 2$. The proof for the case $p > 2$ is similar. Let $\mathcal{C} = \{x \in \mathbb{R}^d \,|\, \|Ax\|_p \le 1\}$. For any $z \notin \mathcal{C}$, define $\mathcal{K}(z) = \{x \in \mathbb{R}^d \,|\, -1 \le g(z)^T x \le 1\}$, where $g(z)$ is a subgradient of $\|Ax\|_p$ at $x = z$. We have $\mathcal{K}(z) \supseteq \mathcal{C}$ and $z \notin \mathcal{K}(z)$, which gives the separation oracle. Let $A = QR_0$ be A's QR factorization. We have

$$\begin{aligned}\|R_0 x\|_2 = \|Ax\|_2 &\le \|Ax\|_p \le n^{1/p-1/2}\|Ax\|_2 \\ &= n^{1/p-1/2}\|R_0 x\|_2 \quad \forall x \in \mathbb{R}^d,\end{aligned}$$

which means $\mathcal{E}_0 = \mathcal{E}(0, R_0^{-1})$ gives an $n^{1/p-1/2}$-rounding of $\mathcal{C}$. Applying Theorem 9, we can find a $2d$-rounding of $\mathcal{C}$ in at most $3.15d^2 \log(n^{1/p-1/2})$ calls to the separation oracle. Let $\mathcal{E} = \mathcal{E}(0, E)$ be the ellipsoid that gives such a rounding. We have

$$\|y\|_2 \le \|AEy\|_p \le 2d\|y\|_2 \quad \forall y \in \mathbb{R}^d.$$

The QR factorization takes $O(nd^2)$ time. Each call to the separation oracle takes $O(nd)$ time. Computing the extreme points of an ellipsoid takes $O(d^3)$ time. In total, we need $O(nd^3 \log n)$ time.

Appendix F. Proof of Theorem 12. The tool we need to verify the FastLpBasis algorithm is simply the equivalence of vector norms. We present the proof for the case $p < 2$. The proof for the case $p > 2$ is similar. Adopt the notation from the FastLpBasis algorithm. G is chosen such that, with a constant probability,

$$\theta_1\|A_i x\|_2 \le \|\tilde{A}_i x\|_2 \le \theta_2\|A_i x\|_2, \quad i = 1, \ldots, N,$$

where $\theta_1 > 0$ and $\theta_2 > 0$ are constants. Conditioning on this event, we have

$$t^{1/p-1/2}/\theta_1 \cdot \tilde{\mathcal{C}} \subseteq \mathcal{C} \subseteq 1/\theta_2 \cdot \tilde{\mathcal{C}}, \tag{17}$$

where $\mathcal{C} = \{x \,|\, \|Ax\|_p \le 1\}$, because for all $x \in \mathbb{R}^d$,

$$\|Ax\|_p^p = \sum_{i=1}^N \|A_i x\|_p^p \le t^{1-p/2} \sum_{i=1}^N \|A_i x\|_2^p \le t^{1-p/2}/\theta_1^p \cdot \sum_{i=1}^N \|\tilde{A}_i x\|_2^p$$

and

$$\|Ax\|_p^p = \sum_{i=1}^N \|A_i x\|_p^p \ge \sum_{i=1}^N \|A_i x\|_2^p \ge 1/\theta_2^p \cdot \sum_{i=1}^N \|\tilde{A}_i x\|_2^p.$$

Let R_0 be the R matrix from the QR decomposition of $\tilde{A}$, and define

$$\mathcal{E}_0 = \{x \,|\, s^{1/p-1/2}\|R_0 x\|_2 \le 1\}.$$

We show that $\mathcal{E}_0$ gives an $(Ns)^{1/p-1/2}$-rounding of $\tilde{C}$. For all $x \in \mathbb{R}^d$, we have

$$\left(\sum_{i=1}^{N} \|\tilde{A}_i x\|_2^p\right)^{1/p} \leq N^{1/p-1/2} \left(\sum_{i=1}^{N} \|\tilde{A}_i x\|_2^2\right)^{1/2} = N^{1/p-1/2} \|R_0 x\|_2$$

and

$$\begin{aligned}\left(\sum_{i=1}^{N} \|\tilde{A}_i x\|_2^p\right)^{1/p} &\geq s^{1/2-1/p} \left(\sum_{i=1}^{N} \|\tilde{A}_i x\|_p^p\right)^{1/p} = s^{1/2-1/p} \|\tilde{A}x\|_p \\ &\geq s^{1/2-1/p} \|\tilde{A}x\|_2 = s^{1/2-1/p} \|R_0 x\|_2.\end{aligned}$$

Hence $\mathcal{E}_0$ gives an $(Ns)^{1/p-1/2}$-rounding of $\tilde{\mathcal{C}}$. Then we compute a $(2d)$-rounding of $\tilde{\mathcal{C}}$ and obtain the matrix R. The running time is $\mathcal{O}(Nsd^3 \log(Ns)) = \mathcal{O}(nd \log n)$ since $Ns = ns/t = n/d^2$. Then, by (17), we know $\kappa_p(AR^{-1}) = \mathcal{O}(dt^{1/p-1/2})$.

Appendix G. Proof of Theorem 3.

Upper bound. First we prove the upper bound. Let U be an ℓ_1 $(d, 1)$-conditioned basis for L (see section 4.1). Therefore, $\|U\|_1 \leq d$, $\|x\|_\infty \leq \|Ux\|_1$ for all $x \in \mathbb{R}^d$, and for any $y \in L$, $y = Ux$ for some x. Let $y \in L$; we have

$$\|Ry\| = \|RUx\|_1 \leq \|RU\|_1 \|x\|_\infty \leq \|RU\|_1 \|Ux\|_1 = \|RU\|_1 \|y\|_1.$$

Thus, it suffices to prove an upper bound on $\|RU\|_1$. $(RU)_{ij} = \sum_k R_{ik} U_{kj}$ is a Cauchy scaled by $\gamma_{ij} = \|U^{(j)}\|$. So $\|RU\|_1$ is a sum of $r_1 d$ scaled, dependent half-Cauchys with sum of scalings $\gamma = \sum_{i,j} \|U^{(j)}\| = r_1 \|U\|_1$. By Lemma 3,

$$\mathbf{Pr}[\|RU\|_1 > t r_1 \|U\|_1] \leq \frac{(\log(r_1 d) + \log t)}{t} (1 + o(1)).$$

It suffices to set $t = O(\frac{1}{p} \log(r_1 d))$ for the RHS to be at least $1 - \delta$. Since $\|U\|_1 \leq d$, with probability at least $1 - \delta$, $\|RU\|_1 = O(\frac{r_1 d}{\delta} \log(r_1 d))$. Multiplying both sides by $C = 4/r_1$ gives the upper bound.

Lower bound. The lower bound is essentially following the proof of the lower bound in Theorem 5 of [24], and so we only provide a sparse sketch of the proof. Consider an arbitrary, fixed y. The product CRy is distributed as a Cauchy random vector whose components are independent and scaled by $C\|y\|_1$. Therefore,

$$\|CRy\|_1 = C\|y\|_1 \sum_{i=1}^{r_1} |X_i|,$$

where X_i are i.i.d. Cauchy random variables. We now apply Lemma 4 with $\gamma = r_1 C \|y\|_1$, $\beta^2 = r_1$ and setting $t = \frac{1}{2}$ to obtain

$$\mathbf{Pr}\left[\|CRy\|_1 \leq \frac{1}{2} r_1 C \|y\|_1\right] \leq \exp(-r_1/12).$$

Since $C = 4/r_1$, we have $\mathbf{Pr}[\|CRy\|_1 \leq 2\|y\|_1] \leq \exp(-r_1/12)$. The result now follows by putting a γ-net Γ on L for sufficiently small γ. This argument follows along the same line as the end of section 3 of [24].

It suffices to show the result for $\|y\|_1 = 1$. Consider the γ-net on L with cubes of side γ/d. There are $(2d/\gamma)^d$ such cubes required to cover the hypercube $\|y\|_\infty \leq 1$;

and, for any two points y_1, y_2 inside the same γ/d-cube, $\|y_1 - y_2\|_1 \leq \gamma$. From each of the γ/d-cubes, select a fixed representative point which we will generically refer to as y^*; select the representative to have $\|y^*\|_1 = 1$ if possible. By a union bound,

$$\mathbf{Pr}\left[\min_{y^*} \|CRy^*\|_1 / \|y^*\|_1 < 2\right] \leq (2d/\gamma)^d \exp(-r_1/12).$$

We will thus condition on the high probability event that $\|CRy^*\|_1 \geq \|y^*\|_1$ for all y^*. We will also condition on the upper bound holding (which is true with probability at least $1-\delta$). For any $y \in L$ with $\|y\|_1 = 1$, let y^* denote the representative point for the cube in which y resides (by construction, $\|y^*\|_1 = 1$ as well). Then, $\|y - y^*\| \leq \gamma$ and $y - y^* \in L$ since $y, y^* \in L$ and L is a subspace. We have

$$\|CRy\|_1 = \|CRy^* + CR(y-y^*)\|_1 \geq \|CRy^*\|_1 - \|CR(y-y^*)\|_1 \geq 2\|y^*\|_1 - \kappa\|y - y^*\|_1,$$

where we used the upper bound in the last inequality $\kappa' = \frac{1}{\delta} \cdot O(d \log(r_1 d))$. By choosing $\gamma = 1/\kappa'$ and recalling that $\|y^*\|_1 = 1$, we have that $\|CRy\|_1 \geq 1$, with probability at least $1 - \delta - \exp(-r_1/12 + d\log(2d\kappa'))$. Recall that $\kappa' = O(\frac{d}{\delta}\log(r_1 d))$, so, for c large enough, by picking $r_1 = c \cdot d \log \frac{d}{\delta}$, we satisfy $\frac{r_1}{12} \geq \log \frac{1}{\delta} + d \log(2\frac{d^2}{\delta}\log(r_1 d))$, and so our bounds hold with probability at least $1 - 2\delta$.

Appendix H. Proof of Lemma 9. We will need some lemmas from prior work. The first two lemmas are on properties of a γ-net, taken directly from Lemma 4 of [3]. Let $U \in \mathbb{R}^{t \times d}$ be a matrix whose columns are an orthonormal basis for L; let S be the unit sphere in $\mathbb{R}^d$, and let T be the set of points in S_L, the intersection of L and S, defined by

$$T = \left\{ w : w \in \frac{\gamma}{\sqrt{d}}\mathbb{Z}^d,\ \|w\|_2 \leq 1 \right\},$$

where $\mathbb{Z}^d$ is the d-dimensional integer lattice on (the orthonormal basis for) L. The set T is a γ-net on S_L because every point in S_L is at most ℓ_2-distance γ from some point in T.

LEMMA 16 (Lemma 4 of [3]). *$|T| \leq e^{cd}$ for $c = (\frac{1}{\gamma} + 2)$.*

LEMMA 17 (Lemma 4 of [3]). *For any $d \times d$ matrix M, if for every $u, v \in T$ we have $|u^T M v| \leq \varepsilon$, then for every unit vector w we have $|w^T M w| \leq \frac{\varepsilon}{(1-\gamma)^2}$.*

Note that as $\gamma \to 0$, the inequality in Lemma 17 gets stronger, but the bound on $|T|$ in Lemma 16 gets larger.

The next lemma demonstrates that a JLP distribution preserves matrix products.

LEMMA 18 (Theorem 19 of [18]). *For $\varepsilon \in (0, \frac{1}{2}]$, let G be an $s \times t$ matrix drawn from an MJLP distribution as given in Definition 2. Then for A, B any real matrices with t rows and $\|A\|_F = \|B\|_F = 1$,*

$$\mathbf{Pr}_G[\|A^T G^T G B - A^T B\|_F > 3\varepsilon/2] < c_1 e^{-c_2 s \varepsilon^2}.$$

We now prove the first part of Lemma 9. Let M be the $d \times d$ matrix $M = U^T G^T G U - I$, and let T be the γ-net with $\gamma = \frac{1}{2}$. By Lemma 16, $|T| \leq e^{4d}$. Let $u, v \in T$ be any two points in T, and set $A = Uu$, $B = Uv$ to be two matrices (actually vectors) with t rows. Since U has orthonormal columns, $\|A\|_F = \|B\|_F = 1$. By Lemma 18, after relabeling $3\varepsilon/2 \to \varepsilon$,

$$\mathbf{Pr}_G[|u^T U^T G^T G U v - u^T v| > \varepsilon] \leq c_1 e^{-4c_2 s \varepsilon^2/9}.$$

So, applying the union bound, for every pair $x, y \in T$,

$$|x^T U^T G G^T U y - x^T y| = |x^T M y| \leq \varepsilon$$

holds with probability at least $1 - c_1 |T|^2 e^{-4c_2 s\varepsilon^2/9}$. Let G be the $s \times t$ MJLP matrix constructed as per Lemma 8. We will now derive a bound on s for the first result (2-norm) to hold. For every unit norm x in L, $x = Uw$ for unit norm $w \in \mathbb{R}^d$. By Lemma 17 (with $\gamma = \frac{1}{2}$), for every unit vector $w \in S_L$,

$$|w^T U^T G^T G U w - \|w\|_2^2| \leq 4\varepsilon.$$

Since $w^T U^T G^T G U w = \|Gx\|_2^2$ and $\|w\|_2^2 = \|x\|_2^2$, after rescaling $4\varepsilon \to \varepsilon$, we have proved that with probability at least $1 - c_1 e^{8d} e^{-c_2 s 4\varepsilon^2/(9\cdot 16)} = 1 - c_1 e^{8d} e^{-c_2 s\varepsilon^2/36}$,

$$\sqrt{1-\varepsilon}\|x\|_2 \leq \|Gx\|_2 \leq \sqrt{1+\varepsilon}\|x\|_2.$$

We now derive the second result (Manhattan norm), conditioning on the high probability event that the result holds for the 2-norm as proved above. Since G is an MJLP, we also have that with probability at least $1 - c_1 |T| e^{-c_2 s\varepsilon^2}$, for every $w \in T$ with $x = Uw$,

$$c_3\sqrt{s}(1-\varepsilon)\|x\|_2 \leq \|Gx\|_1 \leq c_3\sqrt{s}(1+\varepsilon)\|x\|_2. \tag{18}$$

Now consider any unit 2-norm $x \in L$; $x = U(w + \Delta)$, where $w \in T$ has 2-norm at most 1, $w + \Delta$ has unit 2-norm, and $\|\Delta\|_2 \leq \gamma$ because T is a γ-net on S. Then,

$$\|Gx\|_1 = \|GUw + GU\Delta\|_1 = \|GUw\|_1 + \Delta',$$

where $|\Delta'| \leq \|GU\Delta\|_1$. We can bound the first term on the RHS using (18). To bound the second term, use the 2-norm bound as follows:

$$\|GU\Delta\|_1 \leq \sqrt{s}\|GU\Delta\|_2 \leq \sqrt{s(1+\varepsilon)}\|U\Delta\|_2 = \sqrt{s(1+\varepsilon)}\|\Delta\|_2 \leq \sqrt{2s}\gamma$$

(the last inequality is because $\varepsilon \leq 1$). Thus, for every unit norm $x \in L$,

$$c_3\sqrt{s}(1-\varepsilon) - 2\gamma\sqrt{s} \leq \|Gx\|_1 \leq c_3\sqrt{s}(1+\varepsilon) + 2\gamma\sqrt{s}.$$

Choosing $\gamma = c_3\varepsilon/2$, $|T| = \exp\left(2d(1+\frac{1}{c_3\varepsilon\sqrt{2}})\right)$. Since $c_3 < (1+\varepsilon)/(1-\varepsilon)$ (as otherwise, by the two properties of an MJLP, $\|Gx\|_1 > \sqrt{s}\|Gx\|_2$ for some x, a contradiction) and $\varepsilon \leq \frac{1}{3}$, with probability at least $1 - c_1 e^{4d/c_3\varepsilon} e^{-c_2 s\varepsilon^2}$,

$$c_3\sqrt{s}(1-2\varepsilon) \leq \|Gx\|_1 \leq c_3\sqrt{s}(1+2\varepsilon).$$

After rescaling $2\varepsilon \to \varepsilon$, the probability becomes at least $1 - c_1 e^{8d/c_3\varepsilon} e^{-c_2 s\varepsilon^2/4}$. Taking a union bound over the 2-norm result and the Manhattan norm result, and using $8d \leq 8d/c_3\varepsilon$, given that $c_3 \leq (1+\varepsilon)/(1-\varepsilon)$ and $\varepsilon \leq 1/3$, we finally have that for any unit 2-norm x, both the inequalities

$$(1-\varepsilon) \leq \|Gx\|_2 \leq (1+\varepsilon),$$
$$c_3\sqrt{s}(1-\varepsilon) \leq \|Gx\|_1 \leq c_3\sqrt{s}(1+\varepsilon)$$

hold with probability at least $1 - 2c_1 e^{8d/c_3\varepsilon} e^{-c_2 s\varepsilon^2/36} = 1 - e^{-k}$, where the last equality

follows by setting $s = 36(k + \frac{8d}{c_3\varepsilon} + \log(2c_1))/c_2\varepsilon^2 = O(\frac{k}{\varepsilon^2} + \frac{d}{\varepsilon^3})$. Since the result holds for any unit norm x, it holds for any x by scaling by $\|x\|_2$.

REFERENCES

[1] N. Ailon and B. Chazelle, *The fast Johnson–Lindenstrauss transform and approximate nearest neighbors*, SIAM J. Comput., 39 (2009), pp. 302–322, doi:10.1137/060673096.

[2] N. Ailon and E. Liberty, *Fast dimension reduction using Rademacher series on dual BCH codes*, in Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms, 2008, pp. 1–9.

[3] S. Arora, E. Hazan, and S. Kale, *A fast random sampling algorithm for sparsifying matrices*, in Proceedings of the 10th International Workshop on Randomization and Computation, 2006, pp. 272–279.

[4] H. Avron, P. Maymounkov, and S. Toledo, *Blendenpik: Supercharging LAPACK's least-squares solver*, SIAM J. Sci. Comput., 32 (2010), pp. 1217–1236, doi:10.1137/090767911.

[5] J. P. Brooks and J. H. Dulá, *The L_1-norm best-fit hyperplane problem*, Appl. Math. Lett., 26 (2013), pp. 51–55.

[6] K. Clarkson, *Subgradient and sampling algorithms for ℓ_1 regression*, in Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms, 2005, pp. 257–266.

[7] A. Dasgupta, P. Drineas, B. Harb, R. Kumar, and M. W. Mahoney, *Sampling algorithms and coresets for ℓ_p regression*, SIAM J. Comput., 38 (2009), pp. 2060–2078, doi:10.1137/070696507.

[8] D. Donoho and X. Huo, *Uncertainty principles and ideal atomic decomposition*, IEEE Trans. Inform. Theory, 47 (2001), pp. 2845–2862.

[9] D. L. Donoho and P. B. Stark, *Uncertainty principles and signal recovery*, SIAM J. Appl. Math., 49 (1989), pp. 906–931, doi:10.1137/0149053.

[10] P. Drineas, M. Magdon-Ismail, M. W. Mahoney, and D. P. Woodruff, *Fast approximation of matrix coherence and statistical leverage*, in Proceedings of the 29th International Conference on Machine Learning, 2012.

[11] P. Drineas, M. W. Mahoney, and S. Muthukrishnan, *Relative-error CUR matrix decompositions*, SIAM J. Matrix Anal. Appl., 30 (2008), pp. 844–881, doi:10.1137/07070471X.

[12] P. Drineas, M. W. Mahoney, S. Muthukrishnan, and T. Sarlós, *Faster least squares approximation*, Numer. Math., 117 (2010), pp. 219–249.

[13] R. Gribonval and M. Nielsen, *Sparse representations in unions of bases*, IEEE Trans. Inform. Theory, 49 (2003), pp. 3320–3325.

[14] M. Gu and S. C. Eisenstat, *A stable and efficient algorithm for the rank-one modification of the symmetric eigenproblem*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 1266–1276, doi:10.1137/S089547989223924X.

[15] P. Indyk, *Stable distributions, pseudorandom generators, embeddings, and data stream computation*, J. ACM, 53 (2006), pp. 307–323.

[16] P. Indyk, *Uncertainty principles, extractors, and explicit embeddings of ℓ_2 into ℓ_1*, in Proceedings of the 39th Annual ACM Symposium on Theory of Computing, 2007, pp. 615–620.

[17] F. John, *Extremum problems with inequalities as subsidiary conditions*, in Studies and Essays Presented to R. Courant on His 60th Birthday, Interscience, New York, 1948, pp. 187–204.

[18] D. M. Kane and J. Nelson, *Sparser Johnson-Lindenstrauss transforms*, in Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, 2012, pp. 1195–1206, doi:10.1137/1.9781611973099.94.

[19] L. Lovász, *An Algorithmic Theory of Numbers, Graphs and Convexity*, CBMS-NSF Regional Conf. Ser. in Appl. Math. 50, SIAM, Philadelphia, 1986, doi:10.1137/1.9781611970203.

[20] M. W. Mahoney, *Randomized algorithms for matrices and data*, Found. Trends Mach. Learn., 3 (2011), pp. 123–224.

[21] M. W. Mahoney and P. Drineas, *CUR matrix decompositions for improved data analysis*, Proc. Natl. Acad. Sci. USA, 106 (2009), pp. 697–702.

[22] A. Maurer, *A bound on the deviation probability for sums of non-negative random variables*, J. Inequal. Pure Appl. Math., 4 (2003), 15.

[23] P. Paschou, J. Lewis, A. Javed, and P. Drineas, *Ancestry informative markers for fine-scale individual assignment to worldwide populations*, J. Med. Genet., 47 (2010), pp. 835–847, doi:10.1136/jmg.2010.078212.

[24] C. Sohler and D. P. Woodruff, *Subspace embeddings for the ℓ_1-norm with applications*, in Proceedings of the 43rd Annual ACM Symposium on Theory of Computing, 2011, pp. 755–764.

[25] M. J. Todd, *On minimum volume ellipsoids containing part of a given ellipsoid*, Math. Oper. Res., 7 (1982), pp. 253–261.
[26] A. Torralba, R. Fergus, and W. T. Freeman, 80 *million tiny images: A large data set for nonparametric object and scene recognition*, IEEE Trans. Pattern Anal. Mach. Intell., 30 (2008), pp. 1958–1970.
[27] J. A. Tropp, *Topics in Sparse Approximation*, Ph.D. thesis, The University of Texas at Austin, Austin, TX, 2004.
[28] J. A. Tropp, *Improved analysis of the subsampled randomized Hadamard transform*, Adv. Adapt. Data Anal., 3 (2011), pp. 115–126.