Chaining, Group Leverage Score Overestimates, and Fast Spectral Hypergraph Sparsification*

Arun Jambulapati
jmblpati@uw.edu
University of Washington
USA

Yang P. Liu
yangpliu@stanford.edu
Stanford University
USA

Aaron Sidford
sidford@stanford.edu
Stanford University
USA

ABSTRACT

We present an algorithm that given any n-vertex, m-edge, rank r hypergraph constructs a spectral sparsifier with $O(n\varepsilon^{-2}\log n\log r)$ hyperedges in nearly-linear $\widetilde{O}(mr)$ time. This improves in both size and efficiency over a line of work [Bansal-Svensson-Trevisan 2019, Kapralov-Krauthgamer-Tardos-Yoshida 2021] for which the previous best size was $O(\min\{n\varepsilon^{-4}\log^3 n, nr^3\varepsilon^{-2}\log n\})$ and runtime was $\widetilde{O}(mr + n^{O(1)})$.

CCS CONCEPTS

• **Theory of computation → Sparsification and spanners**.

KEYWORDS

hypergraph sparsification, generic chaining, leverage scores

ACM Reference Format:
Arun Jambulapati, Yang P. Liu, and Aaron Sidford. 2023. Chaining, Group Leverage Score Overestimates, and Fast Spectral Hypergraph Sparsification. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing (STOC '23), June 20–23, 2023, Orlando, FL, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3564246.3585136

1 INTRODUCTION

The problem of *sparsification* asks to reduce the size of an object while preserving some desired properties. For example, a *cut sparsifier* reduces the number of edges in a graph while approximately preserving the total weight of each cut, and a *spectral sparsifier* reduces the number of edges in a graph while approximately preserving the spectral form of the Laplacian, or equivalently the electrical energy of any potentials. Over the last few decades, a variety of efficient and effective algorithms have been developed for these notions of graph sparsification [3, 4, 33].

In recent years there has been a variety of work seeking to sparsify more complex objectives (see e.g. [26]). One such example is the problem of spectral hypergraph sparsification (see [31] for discussion), which has seen significant attention. In this setting, formalized by [31], we have a hypergraph $\mathcal{G} = (V, E, v)$, where V denotes a finite vertex set, E denotes the edge set, and $v \in \mathbb{R}^E_{\geq 0}$ denotes non-negative edge weights. Here the edge set is a collection of subsets of V of size at least two, i.e. $E \subseteq \{0,1\}^V$ and $|S| \geq 2$ for all $S \in E$ and G is said to be of rank r if the cardinality of each hyperedge is at most r, i.e. $|S| \leq r$ for all $S \in E$. Consequently, when $r = 2$ a hypergraph is simply an undirected graph. For every vector $x \in \mathbb{R}^V$ we define its associated *energy* in G as

$$f_{\mathcal{G}}(x) \stackrel{\text{def}}{=} \sum_{S\in E} v_S \max_{i,j\in S}(x_i - x_j)^2. \tag{1}$$

The problem of spectral hypergraph sparsification asks to produce a hypergraph $\mathcal{H}$ consisting of a small subset of the hyperedges of $\mathcal{G}$, possibly reweighted, whose energy approximates the energy of $\mathcal{G}$ on all vectors $x \in \mathbb{R}^n$ up to a $(1+\varepsilon)$ multiplicative approximation.

When $r = 2$, spectral hypergraph sparsification exactly reduces to spectral sparsification, where it is known that a random-sampling algorithm can produce a sparsifier with $O(n\varepsilon^{-2}\log n)$ edges [33] (it is known how to improve this bound to $O(n\varepsilon^{-2})$ with more adaptive edge choices [3]). For spectral hypergraph sparsification, a line of work [2, 16, 17] has shown that every hypergraph $\mathcal{G}$ admits a sparsifier with a nearly-linear $O(n\varepsilon^{-4}\log^3 n)$ edges, and is surprisingly independent of the rank r. Additionally, [16] proved that there is a random-sampling algorithm that constructs such a sparsifier with high probability in time $\widetilde{O}(mr + n^{O(1)})$.

Building on this line of work, in particular [16], the main result of this paper is the following Theorem 1.1.

Theorem 1.1 (Hypergraph Sparsification). *There is an algorithm that given a rank r hypergraph $\mathcal{G} = (V, E, v)$ with n vertices computes a $(1+\varepsilon)$-approximate spectral hypergraph sparsifier with $O(n\varepsilon^{-2}\log n\log r)$ hyperedges in nearly-linear time, i.e. $\widetilde{O}(\sum_{S\in E}|S|)$, with high probability in n.*

This result consists of two key ingredients. First, we introduce a broad class of sampling probabilities which we call *group leverage score overestimates* (Definition 1.3). While the sampling weights in [16] took time $\widetilde{O}(mr+n^{O(1)})$ to calculate, we show how to compute our more general weights in nearly-linear, $\widetilde{O}(mr)$, time. Second, we use the generic chaining machinery developed by Talagrand [37] to show that that the sampling algorithm of [16], with group leverage score overestimates, actually produces a $(1+\varepsilon)$-spectral hypergraph sparsifier with $O(n\varepsilon^{-2}\log n\log r)$ edges. This improves over the previous bounds of $O(n\varepsilon^{-4}\log^3 n)$ [16], and $O(n\varepsilon^{-2}r^3\log n)$ [2].

Paper Organization. In the remainder of the introduction we discuss our high level setup required to show Theorem 1.1. After providing notation in Section 1.1, in Section 1.2 we describe a more general matrix formulation of hypergraph sparsification that we work with, which we call a *matrix hypergraph*. In Section 1.3 we then introduce our new definition of group leverage score overestimates

*The full version is available at https://arxiv.org/pdf/2209.10539v1.pdf.

STOC '23, June 20–23, 2023, Orlando, FL, USA

ACM ISBN 978-1-4503-9913-5/23/06…$15.00
https://doi.org/10.1145/3564246.3585136

(Definition 1.3) which can be computed efficiently and still suffices for sampling when constructing sparsifiers. Then, in Section 1.4, we provide a high-level overview of the ideas behind generic chaining, which we use to improve the size bound to $O(n\varepsilon^{-2}\log n\log r)$.

After the introduction, we provide our efficient algorithm for computing group leverage score overestimates in Section 2. In Section 3, we analyze a sampling algorithm that produces a spectral hypergraph sparsifier by using a simplified form of chaining known as Dudley's inequality. The number of hyperedges will be $O(n\varepsilon^{-2}\log^3 n)$. In Section 4 we use the powerful generic chaining machinery presented in [37], specifically the growth functional framework, to improve the hyperedge bound to $O(n\varepsilon^{-2}\log n\log r)$.

1.1 General Notation

Throughout, we use C (or C with a subscript denoting a lemma, theorem, or equation number for clarity) to denote a universal constant. We let $\mathbb{Z}_{\geq\alpha} = \mathbb{Z}\cap[\alpha,+\infty)$ and $\mathbb{R}_{\geq\alpha} = \mathbb{R}\cap[\alpha,+\infty)$. We define $\vec{1}_i$ to be indicator vectors for coordinate i, and let $\vec{0}$ be the zero vector. We let $\mathrm{nnz}(\mathbf{A})$ denote the number of nonzero entries in a matrix $\mathbf{A}$. We use $\dagger$ to denote the Moore-Penrose pseudoinverse of a matrix.. We assume all logs are base e unless otherwise denoted. We say that an algorithm succeeds *with high probability in n* if for any constant $C\geq 1$, there is some choice of constants in the algorithm that makes it have success probability at least $1-n^{-C}$. The reader should think of C as fixed but arbitrary throughout the paper. The constants in our main result Theorem 1.1 will depend on this constant C (see Theorem 3.4).

1.2 A Matrix Generalization of Hypergraph Sparsification

We introduce a generalization of hypergraph sparsification to general matrices that we use throughout the paper. Let $a_1,\ldots,a_m\in\mathbb{R}^n$ denote the rows of a matrix $\mathbf{A}\in\mathbb{R}^{m\times n}$, let $\mathcal{S}=\{S_1,\ldots,S_k\}$ be a partition of $[m]$ into k subsets, so $k=|\mathcal{S}|$, and let each set S_i have a non-negative weight v_i, forming a vector $v\in\mathbb{R}^k$. We denote the tuple of the matrix $\mathbf{A}$, the partition $\mathcal{S}$, and the weights v as the (matrix) hypergraph $\mathcal{G}=(\mathcal{S},\mathbf{A},v)$ (henceforth referred to simply as a *hypergraph*). We define the *rank* of a matrix hypergraph as $r=\max_{S\in\mathcal{S}}|S|$. We will assume $r\geq 2$ throughout, as we can duplicate rows a_i. We let $f_{\mathcal{G}}:\mathbb{R}^d\to\mathbb{R}$ denote the *energy function of* $\mathcal{G}$ where $f_{\mathcal{G}}(x)$, the energy of $\mathcal{G}$ of x, is defined as

$$f_{\mathcal{G}}(x)\stackrel{\text{def}}{=}\sum_{i\in[k]}v_i\max_{j\in S_i}\langle a_j,x\rangle^2. \tag{2}$$

Note that (2) generalizes the hypergraph energy in (1), because for a hypergraph with $n=|V|$ vertices and $k=|E|$ hyperedges, a hyperedge of weight v containing the vertices $F\subseteq V$ can be captured with the vectors $a_i=(\vec{1}_{u_1}-\vec{1}_{u_2})$ for all pairs $u_1,u_2\in F$ with weight v. The rank of the matrix hypergraph will be at most $r(r-1)/2$ if the hypergraph has rank r. By definition, in this case the matrix $\mathbf{A}$ will be the incidence matrix of some multigraph G. We will call matrix hypergraphs where $\mathbf{A}$ comes from a normal hypergraph spectral sparsification instance *graphical hypergraphs*. We will use the term graphical hypergraphs primarily in Theorem 2.6, when we show how to efficiently compute sampling weights for them.

We show that this matrix generalization of hypergraph energy can be sparsified essentially as well as graphical hypergraphs. Here, we say that a matrix hypergraph $\mathcal{H}$ is a $(1+\varepsilon)$-approximate spectral sparsifier of $\mathcal{G}$ if $(1+\varepsilon)^{-1}f_{\mathcal{H}}(x)\leq f_{\mathcal{G}}(x)\leq(1+\varepsilon)f_{\mathcal{H}}(x)$ for all $x\in\mathbb{R}^n$.

Theorem 1.2 (Matrix hypergraph sparsification). *There is an algorithm that given a matrix hypergraph $\mathcal{G}=(\mathcal{S},\mathbf{A},v)$ with $\mathbf{A}\in\mathbb{R}^{m\times n}$, and $r=\max_{S\in\mathcal{S}}|S|$ computes a $(1+\varepsilon)$-approximate spectral hypergraph sparsifier with $O(n\varepsilon^{-2}\log m\log r)$ hyperedges in $\widetilde{O}(\mathrm{nnz}(\mathbf{A})+n^{\omega})$ time.*

Unit matrix hypergraphs: A nice benefit of the general matrix setup is that we may assume that the base hypergraph $\mathcal{G}=(\mathcal{S},\mathbf{A},v)$ has unit weights, i.e. all $v_i=1$. This is without loss of generality, by scaling rows of $\mathbf{A}$, i.e. $\mathbf{A}\leftarrow\mathbf{V}^{1/2}\mathbf{A}$ for $\mathbf{V}=\mathbf{diag}(v)$. We make this assumption for the remainder of the paper, and denote unit matrix hypergraphs as $\mathcal{G}=(\mathcal{S},\mathbf{A})$, omitting the v.

1.3 Group Leverage Score Overestimates

A critical component of the $O(n\varepsilon^{-4}\log^3 n)$ size sparsifier in the previous work was the *balanced weight assignment* [16, Definition 5.1] (elaborated on after Definition 1.3) This was used to prove that the sum of "importances" of the hyperedges was bounded by at most n, generalizing the notion of leverage scores in graphs. In this paper, we introduce a weaker version of a balanced weight assignment, in that we only enforce a one-sided inequality and a total size bound, instead of the substantially tighter condition in [16].

Definition 1.3 (Group Leverage Score Overestimates). We say that $\tau\in\mathbb{R}^{\mathcal{S}}_{\geq 0}$ are *ν-(bounded group leverage score) overestimates* for a unit hypergraph $\mathcal{G}=(\mathcal{S},\mathbf{A})$ with $\mathbf{A}\in\mathbb{R}^{m\times n}$ if $\|\tau\|_1\leq\nu$ and there exist an associated set of weights, $w\in\mathbb{R}^m_{\geq 0}$, such that $\sum_{j\in S_i}w_j=1$ for all $i\in[k]$, and $\max_{j\in S_i}a_j^\top(\mathbf{A}^\top\mathbf{W}\mathbf{A})^\dagger a_j\leq\tau_i$ for all $i\in[k]$ where $\mathbf{W}=\mathbf{diag}(w)$.

Our goal is to give an algorithm which computes group leverage score overestimates ν with $\sum_{i\in[k]}\nu_i=O(n)$. Compared to our Definition 1.3, the balanced weight assignment in [16, Definition 5.1] enforced that for all $j\in S_i$, either $w_j=0$ or $a_j^\top(\mathbf{A}^\top\mathbf{W}\mathbf{A})^\dagger a_j\in[\tau_i/\gamma,\tau_i]$ for a constant $\gamma=O(1)$, without initially enforcing that $\sum_{i\in[k]}\tau_i\leq O(n)$. However, it is not difficult to show that this stronger condition implies that $\sum_{i\in[k]}\tau_i\leq\gamma n$ (see [16, Lemma 6.1]). One reason the balanced weight assignment is a natural definition is that when $\gamma=1$, the weights $w\in\mathbb{R}^m$ producing the assignment are a minimizer of the convex optimization problem

$$\min_{\substack{w\in\mathbb{R}^m_{\geq 0}\\ \sum_{j\in S_i}w_j=1\text{ for all }i\in[k]}}\ -\log\det(\mathbf{A}^\top\mathbf{W}\mathbf{A}).$$

This is essentially the *spanning tree potential* in [16], by the matrix tree theorem.

Nonetheless, we show that the weaker notion in Definition 1.3 still suffices for sampling, as long as $\sum_{i\in[k]}\tau_i\leq O(n)$. Precisely, we analyze the following simple sampling algorithm (variants of which were studied in [16, 33]) where an edge e is kept with probability

p_i defined as

$$p_i = \begin{cases} 1/2 & \text{if } \rho \cdot \tau_i \le 1/2 \\ 1 & \text{otherwise} \end{cases} \tag{3}$$

for an oversampling parameter ρ, and upweighted by a factor of p_i^{-1} so that its value is the same in expectation.

Algorithm 1: `Subsample`$(\mathcal{G} = (\mathcal{S}, \mathbf{A}), \tau \in \mathbb{R}^k_{\ge 0}, \rho)$

input: Rank r unit hypergraph $\mathcal{G} = (\mathcal{S}, \mathbf{A})$, group leverage score overestimates τ (Definition 1.3), and oversampling parameter ρ

Initialize a vector $v \in \mathbb{R}^k$.
for $i \in [k]$ **do**
  $p_i \leftarrow 1/2$ if $\rho \cdot \tau_i \le 1$ and 1 otherwise.
  Set $v'_i \leftarrow p_i^{-1}$ with probability p_i, and 0 otherwise.
end
Return $\mathcal{H} \stackrel{\text{def}}{=} (\mathcal{S}, \mathbf{A}, v')$. `// Can remove all sets` S_i `of` $\mathcal{S}$ `in` $\mathcal{H}$ `where` $v'_i = 0$

To understand why group leverage scores are useful for subsampling, we introduce the following facts which ultimately show that group leverage score overestimates upper bound the maximum contribution of each coordinate $i \in [k]$ to the total energy.

Lemma 1.4. *For any unit hypergraph $\mathcal{G} = (\mathcal{S}, \mathbf{A})$ with $\mathbf{A} \in \mathbb{R}^{m \times n}$ and $w \in \mathbb{R}^m_{\ge 0}$ where $\sum_{j \in S_i} w_j = 1$ for all $i \in [k]$, $x^\top \mathbf{A}^\top \mathbf{W} \mathbf{A} x \le f_{\mathcal{G}}(x)$ for all $x \in \mathbb{R}^n$.*

Proof. Note that $\sum_{j \in S_i} w_j \langle a_j, x \rangle^2 \le \max_{j \in S_i} \langle a_j, x \rangle^2$ for all $i \in [k]$ since $\sum_{j \in S_i} w_j = 1$. Hence

$$x^\top \mathbf{A}^\top \mathbf{W} \mathbf{A} x = \sum_{i \in [k]} \sum_{j \in S_i} w_j \langle a_j, x \rangle^2 \le \sum_{i \in [k]} \max_{j \in S_i} \langle a_j, x \rangle^2 \le f_{\mathcal{G}}(x).$$

□

Lemma 1.5. *For any group leverage scores $\tau \in \mathbb{R}^{\mathcal{S}}_{\ge 0}$ and associated weights $w \in \mathbb{R}^m_{\ge 0}$ for unit hypergraph $\mathcal{G} = (\mathcal{S}, \mathbf{A})$ with $\mathbf{A} \in \mathbb{R}^{m \times n}$, $\max_{j \in S_i} \langle a_j, x \rangle^2 \le \tau_i \cdot x^\top \mathbf{A}^\top \mathbf{W} \mathbf{A} x$ for all $i \in [k]$.*

Proof. We can assume that $x^\top \mathbf{A}^\top \mathbf{W} \mathbf{A} x = 1$ by scaling. Note that

$$\max_{x^\top \mathbf{A}^\top \mathbf{W} \mathbf{A} x = 1} \langle a_j, x \rangle^2 = a_j^\top (\mathbf{A}^\top \mathbf{W} \mathbf{A})^\dagger a_j \le \tau_i$$

for all $j \in S_i$ by Definition 1.3. □

Combining Lemmas 1.4 and 1.5 shows that $\max_{j \in S_i} \langle a_j, x \rangle^2 \le \tau_i f_{\mathcal{G}}(x)$, for all x, i.e. coordinate $i \in [k]$ can only contribute τ_i fraction of the hypergraph energy. Intuitively, this means that sampling proportional to τ_i should produce a sparsifier, though formalizing this intuition and achieving tight bounds is challenging. This is the main goal of Sections 3 and 4.

It is worth remarking on some general connections between the group leverage scores defined in Definition 1.3, and similar notions defined for Lewis weights. In general, there are several settings where iterative/contractive procedures produce weights satisfying a one-sided bound, and where such a bound suffices for applications. Our iterative algorithm for computing group leverage score overestimates (Algorithm 2) is inspired by the algorithm of [8] for computing an approximate John ellipse, corresponding to ℓ_∞ *Lewis weights* [11]. The notion of approximate weights in [8] is very similar to Definition 1.3. Additionally, a one-sided ℓ_p Lewis weight computation sufficed for the algorithm of [14] for ℓ_p regression.

1.4 Overview of Chaining

In the section we introduce the basic intuition behind chaining methods, in particular when applied to analyze our sparsification algorithm which samples by group leverage score overestimates (Algorithm 1). The sampling algorithm proposed in Algorithm 1 keeps a hyperedge $S_i \in \mathcal{S}$ and assigns it weight p_i^{-1} for some probability p_i to produce a hypergraph $\mathcal{H}$. We want to prove that, for an appropriate choice of ρ, the value of $f_{\mathcal{G}}(x)$ is preserved up to a multiplicative $(1+\varepsilon)$ approximation *for all* $x \in \mathbb{R}^n$. Even though it is straightforward to show that $f_{\mathcal{G}}(x)$ is preserved up to $(1+\varepsilon)$-multiplicatively for each *fixed* $x \in \mathbb{R}^n$, there are infinitely many $x \in \mathbb{R}^n$ which prevents us from applying a union bound. Even a naïve discretization leaves exponentially many x to check.

The idea behind chaining is to introduce a sequence of finer and finer ϵ-nets to approximate each x at different scales. Define B as the unit ball of $f_{\mathcal{G}}$, i.e. $B = \{x : f_{\mathcal{G}}(x) \le 1\}$. Consider finite subsets $T_0, T_1, \cdots \subseteq B$ of increasing size, which are our nets. For each $N \ge 0$ let $x_N \in T_N$ be the closest point to x in the metric $d(\cdot, \cdot)$ which we define shortly. Write

$$f_{\mathcal{G}}(x) = f_{\mathcal{G}}(x_0) + \sum_{N \ge 0} f_{\mathcal{G}}(x_{N+1}) - f_{\mathcal{G}}(x_N),$$

where the sum converges because $x_N \to x$. Let $\mathcal{H}$ be the subsampled hypergraph, so we get

$$|f_{\mathcal{G}}(x) - f_{\mathcal{H}}(x)| \le |f_{\mathcal{G}}(x_0) - f_{\mathcal{H}}(x_0)| \tag{4}$$

$$+ \sum_{N \ge 0} |(f_{\mathcal{G}}(x_{N+1}) - f_{\mathcal{G}}(x_N)) - (f_{\mathcal{H}}(x_{N+1}) - f_{\mathcal{H}}(x_N))| \tag{5}$$

by the triangle inequality. Thus we want to bound $|(f_{\mathcal{G}}(y) - f_{\mathcal{G}}(z)) - (f_{\mathcal{H}}(y) - f_{\mathcal{H}}(z))|$ for several pairs (y, z). To analyze this, note that $\mathbb{E}_{\mathcal{H}}[(f_{\mathcal{G}}(y) - f_{\mathcal{G}}(z)) - (f_{\mathcal{H}}(y) - f_{\mathcal{H}}(z))] = 0$ by the definition of $\mathcal{H}$. If we define a *distance*

$$\begin{aligned} d(y, z) &:= \mathrm{Var}_{\mathcal{H}}[f_{\mathcal{H}}(y) - f_{\mathcal{H}}(z)]^{1/2} \\ &= \mathbb{E}_{\mathcal{H}}[((f_{\mathcal{G}}(y) - f_{\mathcal{G}}(z)) - (f_{\mathcal{H}}(y) - f_{\mathcal{H}}(z)))^2]^{1/2}, \end{aligned}$$

by Hoeffding's inequality we know that

$$\begin{aligned} &\Pr[|(f_{\mathcal{G}}(y) - f_{\mathcal{G}}(z)) - (f_{\mathcal{H}}(y) - f_{\mathcal{H}}(z))| \ge \kappa d(y, z)] \\ &\le 2 \exp(-2\kappa^2). \end{aligned}$$

Hence, the probability that for $N \ge 0$, parameter κ_N, and all $x_{N+1} \in T_{N+1}, x_N \in T_N$,

$$|(f_{\mathcal{G}}(x_{N+1}) - f_{\mathcal{G}}(x_N)) - (f_{\mathcal{H}}(x_{N+1}) - f_{\mathcal{H}}(x_N))| \tag{6}$$

$$\le \kappa_N d(x_N, x_{N+1}) \tag{7}$$

is at least $1 - 2|T_N||T_{N+1}| \exp(-\kappa_N^2)$. At this point, up to constants, it makes sense to set $|T_N| = 2^{2^N}$ for all N, and $\kappa_N = C \cdot 2^{N/2}$ for sufficiently large constant C, so that $2|T_N||T_{N+1}| \exp(-\kappa_N^2) \le \exp(-2^{2^N})$. Thus (7) holds for all $N \ge 0$ by a union bound. Plugging this all back into (5) and using that $d(\cdot, \cdot)$ satisfies the triangle inequality (at least up to constants), proves the main chaining theorem, which we formally state in Theorem 3.4.

With the chaining theorem in hand, proving the desired sampling bounds in Theorem 1.1 reduces to constructing sets T_N such that the distances $d(x, T_N) = \min_{y \in T_N} d(x, y)$ are suitably bounded. Surprisingly, the celebrated majorizing measures theorem [13, 34] says in variants of the above setting when the sampling distribution is Gaussian instead of Bernoulli (as in our case), this proof method is optimal, i.e. there exist nets T_N with $|T_N| = 2^{2^N}$ that achieve the true optimal bound. We also believe that in our hypergraph sparsification setting, the Gaussian and Bernoulli sampling processes behave similarly. However, the majorizing measures theorem does not shed light on how to construct the sets T_N. Many previous works on chaining have thus settled for suboptimal bounds such as Dudley's inequality [12], which we use in Section 3 to achieve an $O(n\varepsilon^{-2}\log^3 n)$ bound, or rely on analysis frameworks which require additional structure. Towards achieving a better bound in Section 4, we apply a powerful *growth function* framework of Talagrand which shows how to construct the sets T_N given access to a family of functions satisfying a certain growth condition. We defer a more detailed explanation of our application of the growth function framework and deviations from prior work (in particular, the proof of matrix Chernoff for rank one matrices [28, 37]) to the start of Section 4.

1.5 Related Work

We discuss relevant related work on chaining and sparsification by sampling.

Hypergraph spectral sparsification. Previous works showed that hypergraphs admit sparsifiers with $O(n^3\varepsilon^{-2})$ [31], $O(n\varepsilon^{-2}r^3\log n)$ [2], $O(nr(\varepsilon^{-1}\log n)^{O(1)})$ [17], and finally $O(n\varepsilon^{-4}\log^3 n)$ [16] hyperedges. The independent and concurrent work of Lee [24] also used chaining to show that hypergraphs admit spectral sparsifiers with $O(n\varepsilon^{-2}\log n\log r)$ hyperedges, matching our Theorem 1.1. The result [2] also used chaining methods, however, their chaining was over the space of matrices, instead of vectors as is done in this paper.

Hypergraph cut sparsification. The problem of hypergraph cut sparsification [6, 19] asks to maintain the energy of the hypergraph (see (1)), but only for vectors $x \in \{0,1\}^n$. This generalizes the notion of cut sparsification in graphs. In this setting it is known how to construct hypergraph cut sparsifiers with $O(n\varepsilon^{-2}\log n)$ edges with a random sampling algorithm based on a different notion of "balanced weight assignments" [7]. Their algorithm runs in time $\widetilde{O}(mr + n^{O(1)})$. Because hypergraph spectral sparsification strictly generalizes cut sparsification, our Theorem 1.1 produces a hypergraph cut sparsifier in runtime $\widetilde{O}(mr)$, albeit with $O(n\varepsilon^{-2}\log n\log r)$ hyperedges instead of $O(n\varepsilon^{-2}\log n)$ as shown in [7].

Other sparsification objectives. In general, one can study sparsification of functions $f : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ defined as $f(x) = \sum_{i\in[k]} f_i(x)$. When $f_i(x) = \langle a_i, x\rangle^2$ for a vector $a_i \in \mathbb{R}^n$, this is exactly spectral sparsification of matrices and is now well-understood using tools such as the matrix Chernoff bound. On the contrary, for other functions $f_i(x)$, the best known sparsification results often proceed via chaining methods. Nearly tight (up to logarithmic factors) sparsification results are known for sparsification of ℓ_p norms of matrices, i.e. $f(x) = \|\mathbf{A}x\|_p^p = \sum_{i\in[k]} |\langle a_i, x\rangle|^p$ for all $p \in [0,\infty)$ [5, 29, 30, 35, 36], and the proofs generally rely on combining chaining methods with ℓ_p Lewis weights, a natural importance measure for rows of $\mathbf{A}$ analogous to our group leverage scores (Definition 1.3). For more discussion on ℓ_p norm sparsification, see [11].

Sparsification of several additional convex functions, including Tukey and Huber losses, gamma functions for ℓ_p regression, Orlicz norms, etc., is studied in [26]. The analysis uses chaining methods, among other techniques.

Future work. The authors are optimistic that the methods in [16], this paper, and [36], can provide sparsification results for "ℓ_p hypergraph sparsification" for $p \in [1,2]$, i.e. when the energy function is $f_{\mathcal{G}}(x) \stackrel{\text{def}}{=} \sum_{i\in[k]} \max_{j\in S_i} |\langle a_j, x\rangle|^p$, or even beyond. This paper leaves these questions as an interesting direction for future work.

2 GROUP LEVERAGE SCORE OVERESTIMATES

In this section we provide and analyze efficient algorithms for computing group leverage score overestimates as defined in Definition 1.3. Our principal subroutine is the following Algorithm 2 which turns an algorithm for computing leverage score overestimates for row-reweightings of a matrix $\mathbf{A}$ into *group* leverage score overestimates for a hypergraph induced by $\mathbf{A}$. In this section we introduce leverage scores, their overestimates, and procedures for computing them, introduce and analyze Algorithm 2, and then use these results to compute group leverages score overestimates for matrix hypergraphs and graphical hypergraphs.

First, we introduce leverage scores (Definition 2.1) as well as leverage score overestimates and algorithms for computing them (Definition 2.2).

Definition 2.1 (Leverage scores). For a matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$, the leverage score of row $i \in [m]$ is defined as $\sigma_i(\mathbf{A}) \stackrel{\text{def}}{=} a_i^\top(\mathbf{A}^\top\mathbf{A})^\dagger a_i$. Let $\sigma(\mathbf{A}) \in \mathbb{R}^m$ be the corresponding vector of leverage scores.

It is standard that $\sum_{j\in[m]} \sigma_j(\mathbf{A}) = \text{rank}(\mathbf{A}) \leq n$. Additionally, $\sigma_i(\mathbf{A}) \in [0,1]$ and $\sigma_i(\mathbf{A}) = 0$ if and only if $a_i = \vec{0}$. In all hypergraphs in this paper we assume that it is not the case that $a_i = \vec{0}$ as it would make no contribution to the energy. It is known how to estimate the leverage scores to constant accuracy in $\widetilde{O}(1)$ calls to linear system solvers for $\mathbf{A}^\top\mathbf{D}\mathbf{A}$ for positive diagonal matrices $\mathbf{D}$ (see Theorem 2.3).

In the following definition we overload the term "overestimate" with Definition 1.3 when it is clear if the subject is a matrix or a hypergraph.

Definition 2.2 (Leverage Score Overestimates). We call $\widetilde{\sigma} \in \mathbb{R}^m$, *$\nu$-(bounded leverage score) overestimates for* $\mathbf{A} \in \mathbb{R}^{m\times n}$ if $\|\widetilde{\sigma}\|_1 \leq \nu$ and $\widetilde{\sigma} \geq \sigma(\mathbf{A})$ entrywise. Further, we call a procedure $\mathcal{A}$ a *ν-(bounded leverage score) overestimator for* $\mathbf{A}$ if on input $w \in \mathbb{R}^m_{\geq 0}$ it outputs $\mathcal{A}(w) \in \mathbb{R}^m_{\geq 0}$ which are ν-bounded leverage score overestimates for $\sigma_{\mathbf{A}}(w) \stackrel{\text{def}}{=} \sigma(\mathbf{W}^{1/2}\mathbf{A})$ where $\mathbf{W} \stackrel{\text{def}}{=} \mathbf{diag}(w)$.

Leverage score overestimates have played a prominent role in sparsifcation and linear system solving [9, 15, 18, 20–23, 27, 32]. Our

choice of notation in Definition 2.2 is strongly influenced by these works. Further, there are known efficient algorithms for computing leverage score overestimates in general and faster algorithms in the case of graphs as summarized in the following Theorem 2.3.

Theorem 2.3 (Leverage score approximation, [10, 25, 33]). *There is an algorithm that given a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ produces $O(n)$-overestimates of $\mathbf{A}$ in $\widetilde{O}(\mathrm{nnz}(\mathbf{A}) + n^{\omega})$ time with high probability in n. If $\mathbf{A}$ is additionally the weighted incidence matrix of a graph, i.e. every row i is all zero except for a single w_i and a single $-w_i$ for $w_i \neq 0$, then the runtime improves to $\widetilde{O}(\mathrm{nnz}(\mathbf{A}))$.*

Proof. In both cases, the cited works compute $\widehat{\sigma} \in \mathbb{R}^m_{\geq 0}$ with $\widehat{\sigma}_j \in [(1-\delta)\sigma_j(\mathbf{A}), (1+\delta)\sigma_j(\mathbf{A})]$ with high probability in n for any $\delta > 0$ in the stated runtimes multiplied by $O(\mathrm{poly}(1/\delta))$. Since $\|\sigma(\mathbf{A})\|_1 = \mathrm{rank}(\mathbf{A}) \leq n$ the result follows by invoking these algorithms for constant $\delta > 0$ and outputting $(1-\delta)^{-1}\widehat{\sigma}$. □

Given Theorem 2.3, it suffices to provide an algorithm which carefully combines $\widetilde{O}(1)$ overestimates for matrices to produce overestimates for hypergraphs. We provide an algorithm which does this in Algorithm 2. This algorithm is a natural generalization of the algorithm of [8] for computing an approximate John ellipse mentioned in Section 1.3. The procedure simply iterates on a weight vector $w^{(t)}$, computing $\widetilde{\sigma}^{(t)}$ as leverage score overestimates for $\sigma_{\mathbf{A}}(w)$ (Line 2), and then letting $w^{(t+1)}$ be the natural re-normalization of those weights (Line 4). The procedure then outputs the average of these weights (Line 7) as the weights associated with an overestimate $\tau \in \mathbb{R}^k_{\geq 0}$ where each entry of τ is an appropriately scaled up aggregation of the computed leverage score overestimates (Line 6).

Algorithm 2: `GroupLeverageOverestimate`($\mathcal{G} = (\mathcal{S}, \mathbf{A}), T, \mathcal{A}$)

input: Rank r unit hypergraph $\mathcal{G} = (\mathcal{S}, \mathbf{A})$ with $\mathbf{A} \in \mathbb{R}^{m \times n}$, iteration count $T \in \mathbb{Z}_{\geq 1}$, and ν-overestimator $\mathcal{A}$ for $\mathbf{A}$ (Definition 2.2)

1 Initialize $w^{(1)} \in \mathbb{R}^m_{\geq 0}$ with $w^{(1)}_j = 1/|S_i|$ for all $i \in [k]$ and $j \in S_i$;

2 **for** $t = 1$ *to* T **do**

3 $\quad \widetilde{\sigma}^{(t)} \leftarrow \mathcal{A}(w^{(t)})$; // $\widetilde{\sigma}^{(t)} \in \mathbb{R}^m_{\geq 0}$ with $\|\widetilde{\sigma}^{(t)}\|_1 \leq \nu$ and $\widetilde{\sigma}^{(t)} \geq \sigma_{\mathbf{A}}(w^{(t)})$ entrywise

4 $\quad$ Set $w^{(t+1)} \in \mathbb{R}^m_{>0}$ with $w^{(t+1)}_j \leftarrow \widetilde{\sigma}^{(t)}_j / (\sum_{j' \in S_i} \widetilde{\sigma}^{(t)}_{j'})$ for all $i \in [k]$ and $j \in S_i$;

5 **end**

6 Set $\tau \in \mathbb{R}^k_{\geq 0}$ with $\tau_i \leftarrow \exp(T^{-1}\log r) \cdot \frac{1}{T} \sum_{t \in [T]} \sum_{j \in S_i} \widetilde{\sigma}^{(t)}_j$ for all $i \in [k]$;

7 $\overline{w} \leftarrow \frac{1}{T} \sum_{t \in [T]} w^{(t)}$;

8 **return** $(\tau, \overline{w})$;

For intuition behind this algorithm, consider the optimal weights w^* and group leverage scores τ^*, corresponding to $\gamma = 1$ as discussed in Section 1.3. Precisely, for the hypergraph $\mathcal{G} = (\mathcal{S}, \mathbf{A})$ we have that $a_j^\top (\mathbf{A}^\top \mathbf{W}^* \mathbf{A})^\dagger a_j = \tau^*_i$ for all $i \in [k]$ and $j \in S_i$, unless $w_j = 0$. This can be more compactly written as $[\sigma_{\mathbf{A}}(w^*)]_j = w^*_j \tau^*_i$ for all $i \in [k]$ and $j \in S_i$. Because $\sum_{j \in S_i} w^*_j = 1$, we know that $\tau^*_i = \sum_{j' \in S_i} [\sigma_{\mathbf{A}}(w^*)]_{j'}$ and therefore

$$w^*_j = [\sigma_{\mathbf{A}}(w^*)]_j / \sum_{j' \in S_i} [\sigma_{\mathbf{A}}(w^*)]_{j'}$$

for all $i \in [k]$ and $j \in S_i$. Thus, Algorithm 2 can be viewed as simply updating $w^{(t)}$ as if the above equation was an equality, using overestimates for leverage score, and then averaging the weights over all $t \in [T]$.

In Theorem 2.4 we prove that this algorithm does successfully compute leverage score overestimates. In fact, the theorem implies that it suffices to compute $O(n)$-bounded leverage score overestimates of $O(\log r)$ different reweightings of $\mathbf{A} \in \mathbb{R}^{m \times n}$ in order to compute $O(n)$-bounded group leverage score overestimates of a rank r hypergraph associated with $\mathbf{A}$. The proof is similar to that of [8] for computing approximate John ellipses and uses a critical technical tool of it, the convexity of $\log([\sigma_{\mathbf{A}}(w)]_j / w_j)$ with respect to w for any j.

We note that is not actually clear that $\tau \geq \tau^*$ where τ are the overestimates produced by Algorithm 2 for $\mathcal{G}$ and τ^* are the optimal group leverage scores discussed earlier. It is an interesting open problem to determine whether or not this is the case and if it is false, the term "group leverage score overestimates" is perhaps a misnomer. However, in either case the overestimates produced are sufficient for hypergraph spectral sparsfication as we prove in Sections 3 and 4.

Theorem 2.4 (Group Leverage Score Overestimation Algorithm). *Given any rank r unit hypergraph $\mathcal{G} = (\mathcal{S}, \mathbf{A})$ with $\mathbf{A} \in \mathbb{R}^{m \times n}$, $T \in \mathbb{Z}_{\geq 1}$, and ν-overestimator $\mathcal{A}$ for $\mathbf{A}$ (Definition 2.2),* `GroupLeverageOverestimate`$(\mathcal{G}, T, \nu)$ *in Algorithm 2 outputs $\exp(T^{-1}\log r)\nu$-overestimates $\tau \in \mathbb{R}^{\mathcal{S}}_{>0}$ for $\mathcal{G}$ and associated weights $\overline{w} \in \mathbb{R}^m_{\geq 0}$. The algorithm can be implemented in $O(mT)$ time plus the time of invoking $\mathcal{A}$ on T different inputs.*

Proof. The runtime is immediate from the pseudocode (there are T iterations each of which takes time $O(m)$ plus the time to invoke $\mathcal{A}$) and consequently it suffices to show that τ are $\exp(T^{-1}\log r)\nu$-overestimates for $\mathcal{G}$ with associated weights $\overline{w} \in \mathbb{R}^m_{\geq 0}$. By the definition of τ (Line 6) and $\widetilde{\sigma}$ (Line 2 and Definition 2.2) it follows that

$$\begin{aligned} \|\tau\|_1 &= \exp(T^{-1}\log r) \cdot \sum_{i \in [k]} \left[\frac{1}{T} \sum_{t \in [T]} \sum_{j \in S_i} \widetilde{\sigma}^{(t)}_j \right] \\ &= \frac{\exp(T^{-1}\log r)}{T} \sum_{t \in [T]} \|\widetilde{\sigma}^{(t)}\|_1 \\ &\leq \exp(T^{-1}\log r)\nu \,. \end{aligned}$$

Next, for any $i \in [k]$ and $j \in S_i$ since $\log([\sigma_{\mathbf{A}}(w)]_j / w_j)$ is convex in w [8, Lemma 3.4] it follows that

$$\log\left(\frac{[\sigma_{\mathbf{A}}(\overline{w})]_j}{\overline{w}_j}\right) \leq \frac{1}{T} \sum_{t \in [T]} \log\left(\frac{[\sigma_{\mathbf{A}}(w^{(t)})]_j}{w^{(t)}_j}\right) \quad \text{(convexity [8, Lemma 3.4])}$$

$$\leq \frac{1}{T}\sum_{t\in[T]}\log\left(\frac{\widetilde{\sigma}_j^{(t)}}{w_j^{(t)}}\right)$$
(Definition of $\widetilde{\sigma}$ (Line 2 and Definition 2.2))

$$= \frac{1}{T}\sum_{t\in[T]}\left[\log\left(\frac{w_j^{(t+1)}}{w_j^{(t)}}\right)+\log\left(\sum_{j'\in S_i}\widetilde{\sigma}_{j'}^{(t)}\right)\right]$$

$$\leq \frac{1}{T}\log\left(\frac{w_j^{(T+1)}}{w_j^{(1)}}\right)+\log\left(\frac{1}{T}\sum_{t\in[T]}\sum_{j'\in S_i}\widetilde{\sigma}_{j'}^{(t)}\right)$$
(concavity of $\log(\cdot)$)

$$= \frac{1}{T}\log\left(\frac{w_j^{(T+1)}}{w_j^{(1)}}\right)-\frac{1}{T}\log(r)+\log(\tau_i)\,.$$
(Definition of τ (Line 6))

Now observe that $w_j^{(T)} \leq 1$ (since leverage scores are at most 1) and $w_j^{(1)} = \frac{1}{|S_j|} \geq \frac{1}{r}$ (by definition of $w_j^{(1)}$ and r). Thus $w_j^{(T+1)} \leq r \cdot w_j^{(1)}$ and we have the desired bound as

$$\tau_i \geq \frac{[\sigma_{\mathbf{A}}(\overline{w})]_j}{\overline{w}_j} = a_j^\top(\mathbf{A}\overline{\mathbf{W}}\mathbf{A})^\dagger a_j \text{ where } \overline{\mathbf{W}} = \mathbf{diag}(\overline{w})\,. \qquad \square$$

As an immediate consequence of Theorems 2.3 and 2.4 we obtain an efficient algorithm for computing group leverage score overestimates for general hypergraphs.

Theorem 2.5 (Efficient Overestimates of General Hypergraphs). *There is an algorithm which given any rank r unit hypergraph $\mathcal{G} = (\mathcal{S}, \mathbf{A})$ with $\mathbf{A} \in \mathbb{R}^{m\times n}$ in time $\widetilde{O}(\mathrm{nnz}(\mathbf{A})+n^\omega)$ computes $O(n)$-overestimates for $\mathcal{G}$ with high probability in n.*

Proof. Apply Theorem 2.4 with $T = \Theta(\log r)$ using Theorem 2.3 to efficiently implement the $O(n)$-overestimator. $\square$

Finally we show how to use Theorems 2.3 and 2.4 to obtain an efficient algorithm for computing group leverage score overestimates for graphical hypergraphs. Naïvely applying these results would yield an algorithm that in $\widetilde{O}(\sum_{i\in[k]}|S_i|^2)$ computes $O(n)$-overestimates for an n-node hypergraph with hyperedges $S_1, ..., S_k$. In the following theorem we show how to improve this to $\widetilde{O}(\sum_{i\in[k]}|S_i|)$ using the trick of using stars to overestimate hyperedges [16, Section 3].

Theorem 2.6 (Efficient Overestimates of Graphical Hypergraphs). *There is an an algorithm that given any n-node graphical hypergraph $\mathcal{G} = (V, E, v)$ in time $\widetilde{O}(\sum_{S_i\in E}|S_i|)$ outputs with high probability in n, $O(n)$-overestimates for the matrix unit-hypergraph associated with $\mathcal{G}$.*

Proof. Note that the the matrix unit-hypergraph associated with $\mathcal{G}$, is $(\mathcal{S}, \mathbf{A})$ where $\mathbf{A} \in \mathbb{R}^{m\times V}$ where $m = \sum_{S_i\in E}\binom{|S_i|}{2}$ and each $a, b \in S_i$ with $a \neq b$ has an associated row in $\mathbf{A}$, which we call $j_{a,b,i}$, that is $\sqrt{v_i}(\vec{1}_a - \vec{1}_b)$. Further, each $S_i \in E$ corresponds to a $S_i \in \mathcal{S}$ containing $j_{a,b,i}$ for each $a, b \in S_i$ with $a \neq b$.

Now, for each $S_i \in E$ fix an arbitrary vertex $a_i \in S_i$. Further, consider the unit hypergraph $(\mathcal{S}', \mathbf{A}')$ that consists of discarding from $\mathcal{G}$ the rows $j_{a,b,i}$ where it is not the case that $a = a_i$ and $b \neq a_i$. Note that $\mathbf{A}' \in \mathbb{R}^{m'\times V}$ with $m' = \sum_{S_i\in E}(|S_i|-1)$ and $\mathbf{A}'$ is a weighted incidence matrix of a graph. Consequently, using Theorem 2.4 and Theorem 2.3 we can compute $\tau \in \mathbb{R}^k$ that are $O(n)$-overestimates for $(\mathcal{S}', \mathbf{A}')$ with associated weights $w' \in \mathbb{R}^{m'}_{\geq 0}$ with high probability in n.

Consequently, to complete the proof it suffices to show that $\tau = 2\tau'$ are $O(n)$-overestimates for $(\mathcal{S}, \mathcal{A})$. Clearly $\|\tau\|_1 = 2\|\tau'\|_1 \leq O(n)$ and consequently it suffices to produce associated weights $w \in \mathbb{R}^m_{\geq 0}$. Define such a $w \in \mathbb{R}^m_{\geq 0}$ by setting w_j to have the value of the associated entry in w'_j if row j is in both $\mathbf{A}$ and $\mathbf{A}'$ and 0 otherwise. Since w' were the weights associated with τ', and the only new weights in w are 0, we clearly have the property that for all $S_i \in \mathcal{S}$

$$\sum_{j_{i,a,b}:a,b\in S_i \text{ with } a\neq b} w_{j_{i,a,b}} = 1.$$

The result then follows from the next equation, where $\mathbf{W} \stackrel{\text{def}}{=} \mathbf{diag}(w)$ and $\mathbf{W}' \stackrel{\text{def}}{=} \mathbf{diag}(w')$, and $j_{i,a,b} \in S_i$:

$$\begin{aligned} a_{j_{i,a,b}}^\top(\mathbf{A}^\top\mathbf{W}\mathbf{A})^\dagger a_{j_{i,a,b}} &\stackrel{(i)}{=} v_i(\vec{1}_a - \vec{1}_b)((\mathbf{A}')^\top\mathbf{W}'\mathbf{A}')^\dagger(\vec{1}_a - \vec{1}_b) \\ &\stackrel{(ii)}{\leq} v_i(\vec{1}_a - \vec{1}_{a_i})((\mathbf{A}')^\top\mathbf{W}'\mathbf{A}')^\dagger(\vec{1}_a - \vec{1}_{a_i}) \\ &\quad + v_i(\vec{1}_{a_i} - \vec{1}_b)((\mathbf{A}')^\top\mathbf{W}'\mathbf{A}')^\dagger(\vec{1}_{a_i} - \vec{1}_b) \\ &\stackrel{(iii)}{\leq} \tau'_i + \tau'_i = \tau \end{aligned}$$

Here, (i) follows from the definition of $a_{j_{i,a,b}}$ and w', and (iii) follows because τ' are overestimates for $\mathbf{A}'$ with weights w'. (ii) follows from the triangle inequality for effective resistances in graphs (see [38]). It is worth remarking that if instead set $\tau_i = 4\tau'_i$ (so $\|\tau\|_1 \leq O(\|\tau'\|_1)$ still) that we can simply use the triangle inequality for norms in this line. $\square$

3 SIZE BOUND FROM DUDLEY'S INEQUALITY

In this section, we prove that sampling hyperedges S_i by probabilities p_i produces a sparsifier with high probability. As a warmup to the results in the following Section 4, we first prove `Subsample` achieves a weaker size bound of $3k/4 + O(n\varepsilon^{-2}\log^3 m)$ using a simple form of chaining. (We discuss how to iterate this bound to $O(n\varepsilon^{-2}\log^3(n/\varepsilon))$ at the end of this section.) In the context of previous work on chaining, our proof is essentially just applying Dudley's entropy bound [12] instead of the full generic chaining (we elaborate on this after Theorem 3.4). Because the proof is relatively simple and provides nice intuition for the more complicated analysis in Section 4, we give a self-contained analysis except for an ℓ_∞ ball covering theorem from [1].

Specifically, we prove the following theorem.

Theorem 3.1. *Let $\mathcal{G} = (\mathcal{S}, \mathbf{A})$ be a unit hypergraph with k hyperedges, and let τ be given group leverage scores (Definition 1.3) with valid weights w (which do not need to be known). For any constant C, there is an absolute constant C_1 (depending on C) such that $\mathcal{H} = \mathtt{Subsample}(\mathcal{G}, \tau, \rho)$ (Algorithm 1) with $\rho = C_1\varepsilon^{-2}\log^3 m$ satisfies with probability at least $1 - n^{-C}$ that*

$$(1-\varepsilon)f_{\mathcal{G}}(x) \leq f_{\mathcal{H}}(x) \leq (1+\varepsilon)f_{\mathcal{H}}(x) \text{ for all } x \in \mathbb{R}^n$$

and has at most $3k/4 + O(n\varepsilon^{-2}\log^3 m)$ edges.

If τ is given by Theorem 2.4, the above applied to an k-edge hypergraph gives a sparsifier with $3k/4 + O(n\varepsilon^{-2}\log^3 m)$. We will later improve this bound to $O(n\varepsilon^{-2}\log m \log r)$ in Section 4.

Let us discuss our general proof strategy for Theorem 3.1. In a chaining argument, it is useful to study how the *difference* between energies of two points $x, y \in \mathbb{R}^n$, i.e. $f_{\mathcal{G}}(x) - f_{\mathcal{G}}(y)$, is affected by sampling. By construction, the sampling is unbiased for any fixed input $x \in \mathbb{R}^n$, so $\mathbb{E}_{\mathcal{H}}[f_{\mathcal{H}}(x) - f_{\mathcal{H}}(y)] = f_{\mathcal{G}}(x) - f_{\mathcal{G}}(y)$. As is standard in chaining setups, we now define a distance function which is an upper bound on the variance.

Definition 3.2 (Metric Space). For a fixed hypergraph $\mathcal{G} = (\mathcal{S}, \mathbf{A})$, define $g_i(x) \stackrel{\text{def}}{=} \max_{j \in S_i} |\langle a_j, x\rangle|$ for all $S_i \in \mathcal{S}$. We let B be the unit ball of the energy function, i.e. $B \stackrel{\text{def}}{=} \{f_{\mathcal{G}}(x) \le 1 : x \in \mathbb{R}^n\}$. Additionally, for given sampling probabilities $p_i \in \{1/2, 1\}$, we define the *distance function* $d : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ for all $x, y \in \mathbb{R}^n$ as

$$d(x,y) \stackrel{\text{def}}{=} \left(\sum_{i\in[k]} \mathbf{1}_{\{p_i \neq 1\}} p_i^{-1}(g_i(x)^2 - g_i(y)^2)^2\right)^{1/2}. \tag{8}$$

Further, for a finite subset $T \subseteq \mathbb{R}^n$, we define $d(x,T) \stackrel{\text{def}}{=} \min_{t\in T} d(x,t)$.

We observe that the functions g_i are convex and satisfy $f_{\mathcal{G}}(x) = \sum_{i\in[k]} g_i(x)^2$. We formalize additional key properties of the distance function d in the following lemma.

Lemma 3.3. *Let $\mathcal{G} = (\mathcal{S}, \mathbf{A})$ be a hypergraph, let $p_i \in (0,1]$ be given, and let $d(\cdot,\cdot)$ be defined as in Definition 3.2. Let $\mathcal{H} = (\mathcal{S}, \mathbf{A}, \widehat{v})$, where $\widehat{v} \in \mathbb{R}^k$ is defined as $\widehat{v}_i = p_i^{-1}$ with probability p_i and 0 otherwise. d satisfies the following properties for any $x, y, z \in \mathbb{R}^n$:*

- $\mathrm{Var}_{\mathcal{H}}[f_{\mathcal{H}}(x) - f_{\mathcal{H}}(y)] \le d(x,y)^2$
- $d(x,z) \le d(x,y) + d(y,z)$.

Proof. To bound the variance, note that $f_{\mathcal{H}}(x) - f_{\mathcal{H}}(y)$ is a sum of k independent random variables, where the i^{th} variable is either 0 or $p_i^{-1}(g_i(x)^2 - g_i(y)^2)$ if $p_i \neq 1$ and always $(g_i(x)^2 - g_i(y)^2)$ otherwise. Thus we have

$$\begin{aligned}\mathrm{Var}[f_{\mathcal{H}}(x) - f_{\mathcal{H}}(y)] &= \sum_{i\in[k]} \frac{1-p_i}{p_i}(g_i(x)^2 - g_i(y)^2)^2 \\ &\le d(x,y)^2\end{aligned}$$

as $p_i \in (0,1]$. This shows the first property. For the second property, define $v^x \in \mathbb{R}^k$ as the vector with coordinates

$$v_i^x \stackrel{\text{def}}{=} \sqrt{\mathbf{1}_{\{p_i\neq 1\}} p_i^{-1}}\, g_i(x)^2$$

for all $i \in [m]$. Define v^y, v^z similarly. Now the desired bound of the final property follows because $d(x,y) = \|v^x - v^y\|_2$, and by triangle inequality

$$d(x,z) = \|v^x - v^z\|_2 \le \|v^x - v^y\|_2 + \|v^y - v^z\|_2 = d(x,y) + d(y,z). \quad \square$$

With these facts in hand, we describe our formal chaining setup.

Theorem 3.4 (Chaining). *Define $\mathcal{G}$, p_i, d, $\mathcal{H}$ as in Lemma 3.3. For $s \ge \lceil \log_2(\log n)\rceil$, define*

$$\gamma \stackrel{\text{def}}{=} \inf_{\substack{T_s, T_{s+1}, \ldots \\ T_N \subseteq B, |T_N| \le 2^{2^N} \text{ for all } N \ge s}} \sup_{x\in B} 2^{s/2}\cdot d(x,\vec{0}) + \sum_{N\ge s} 2^{N/2} d(x, T_N). \tag{9}$$

Then there is an absolute constant C_2 such that with high probability (i.e. at least $1 - n^{-C}$, and C_2 depends on C) for all $x \in \mathbb{R}^n$

$$(1 - C_2\gamma) f_{\mathcal{G}}(x) \le f_{\mathcal{H}}(x) \le (1 + C_2\gamma) f_{\mathcal{H}}(x).$$

To prove Theorem 3.4 we first note the following simple application of Hoeffding's inequality.

Lemma 3.5. *For any subsets $X, Y \subseteq \mathbb{R}^n$ and $K \ge 0$ we have that with probability at least $1 - 2|X||Y|\exp(-2K^2)$ over choices of $\mathcal{H}$ that for all $x \in X, y \in Y$ that $|(f_{\mathcal{G}}(x) - f_{\mathcal{G}}(y)) - (f_{\mathcal{H}}(x) - f_{\mathcal{H}}(y))| \le K \cdot d(x,y)$.*

Proof. Note that for each pair $x \in X, y \in Y$ we have that $\mathbb{E}[f_{\mathcal{H}}(x) - f_{\mathcal{H}}(y)] = f_{\mathcal{G}}(x) - f_{\mathcal{G}}(y)$ and that $f_{\mathcal{H}}(x) - f_{\mathcal{H}}(y)$ is a sum of k independent random variables, where the i^{th} variable is either 0 or $p_i^{-1}(g_i(x)^2 - g_i(y)^2)$ if $p_i \neq 1$ and always $(g_i(x)^2 - g_i(y)^2)$ otherwise by Lemma 3.3. Applying Hoeffding's inequality, the definition of d (see Lemma 3.3), and the fact that $p_i \in \{1/2, 1\}$ yields

$$\begin{aligned}&\Pr_{\mathcal{H}}\left[|(f_{\mathcal{G}}(x) - f_{\mathcal{G}}(y)) - (f_{\mathcal{H}}(x) - f_{\mathcal{H}}(y))| > K\cdot d(x,y)\right] \\ &= \Pr_{\mathcal{H}}\left[|\mathbb{E}[f_{\mathcal{H}}(x) - f_{\mathcal{H}}(y)] - (f_{\mathcal{H}}(x) - f_{\mathcal{H}}(y))| > K\cdot d(x,y)\right] \\ &\le 2\exp\left(-\frac{2K^2\cdot d(x,y)^2}{\sum_{i\in[k]} \mathbf{1}_{\{p_i\neq 1\}} p_i^{-1}(g_i(x)^2 - g_i(y)^2)^2}\right) \\ &= 2\exp\left(-\frac{2K^2\cdot d(x,y)^2}{d(x,y)^2}\right) = 2\exp(-2K^2).\end{aligned}$$

The claim follows by union bounding over all $x \in X, y \in Y$. $\square$

To prove Theorem 3.4 we apply Lemma 3.5 on all levels $N \ge 0$ and add them up.

Proof of Theorem 3.4. Consider the event E_N that for all $x \in T_N, y \in T_{N+1}$

$$|(f_{\mathcal{G}}(x) - f_{\mathcal{G}}(y)) - (f_{\mathcal{H}}(x) - f_{\mathcal{H}}(y))| \le 2\sqrt{C}\cdot 2^{N/2} d(x,y).$$

We claim that $\Pr_{\mathcal{H}}[E_N] \ge 1 - 1/2\cdot n^{-C} 2^{-2^N}$. Indeed this is true by taking $X = T_N$, $Y = T_{N+1}$, and $K = 2\sqrt{C}\cdot 2^{N/2}$ in Lemma 3.5 and noting that

$$2|T_N||T_{N+1}|\exp(-2K^2) \le 2\cdot 2^{3\cdot 2^N}\exp(-8C\cdot 2^N) \le 1/2\cdot n^{-C}2^{-2^N}$$

because we assume $C \ge 1$ and $2^N \ge 2^s \ge \log n$. Hence all events $E_s, E_{s+1}, \ldots$ hold with probability at least $1 - \sum_{N\ge 0} 1/2\cdot n^{-C}2^{-2^N} \ge 1 - 1/2\cdot n^{-C}$.

By setting $X = T_s$, $Y = \{0\}$, $K = 2\sqrt{C}\cdot 2^{s/2}$ and again applying Lemma 3.5 we get that

$$|f_{\mathcal{G}}(x) - f_{\mathcal{H}}(x)| \le 2\sqrt{C}\cdot 2^{s/2}\cdot d(x,\vec{0}) \text{ for all } x \in T_N \tag{10}$$

for all $x \in T_s$ with probability at least

$$1 - |T_s|\exp(-8C\cdot 2^s) \ge 1 - 1/2\cdot n^{-C}$$

because we assume $C \ge 1$ and $2^s \ge \log n$. Hence all events $E_s, E_{s+1}, \ldots$ and (10) hold with probability at least $1 - n^{-C}$. Now, for each $x \in B$

let $x_N = \operatorname{argmin}_{y \in T_N} d(x, y)$. If all events above hold, then

$$\begin{aligned}
&|f_{\mathcal{G}}(x) - f_{\mathcal{H}}(x)| \le |f_{\mathcal{G}}(x_s) - f_{\mathcal{H}}(x_s)| \\
&+ \sum_{N \ge s} |f_{\mathcal{G}}(x_N) - f_{\mathcal{G}}(x_{N+1}) - (f_{\mathcal{H}}(x_N) - f_{\mathcal{H}}(x_{N+1}))| \\
&\le 2\sqrt{C} \cdot 2^{s/2} \cdot d(x_s, 0) + 2\sqrt{C} \sum_{N \ge s} 2^{N/2} d(x_N, x_{N+1}) \\
&\overset{(i)}{\le} 2\sqrt{C} \cdot 2^{s/2} \cdot (d(x_s, x) + d(x, \vec{0})) \\
&+ 2\sqrt{C} \sum_{N \ge s} 2^{N/2} (d(x, x_N) + d(x, x_{N+1})) \\
&\le 2\sqrt{C} \cdot 2^{s/2} \cdot d(x, \vec{0}) + 2\sqrt{C} \sum_{N \ge s} \left(2^{(N-1)/2} + 2^{N/2}\right) d(x, T_N) \\
&\le 4\sqrt{C} \cdot 2^{s/2} \cdot d(x, \vec{0}) + 4\sqrt{C} \sum_{N \ge s} 2^{N/2} d(x, T_N) \le 4\sqrt{C}\gamma,
\end{aligned}$$

where (i) uses that d is a metric (Lemma 3.3). Thus we may set $C_2 = 4\sqrt{C}$. □

The goal of the remainder of this section is to bound the quantity in (9) for sampling probabilities p_i given by (3), where ρ is an oversampling parameter. In this section, we will set $\rho = C_1 \varepsilon^{-2} \log^3 m$: later in Section 4 we modify the chaining argument to show $\rho = C\varepsilon^{-2} \log m \log r$ still suffices for some sufficiently large constant C. Because $\sum_{i \in [k]} \tau_i \le O(n)$ by Theorem 2.4, the hypergraph $\mathcal{H}$ will have $3k/4 + O(\rho n)$ edges with high probability.

We first handle the term $2^{s/2} \cdot d(x, \vec{0})$ in Theorem 3.4. This calculation provides critical intuition for why group leverage scores are sufficient for sampling.

Lemma 3.6 (Handling $d(x, \vec{0})$). *For group leverage score overestimates τ and corresponding weights w (Definition 1.3), $x \in B$, and p_i given by (3) we have $d(x, \vec{0}) \le \rho^{-1/2}$.*

Proof. Note that $\mathbf{1}_{p_i \ne 1} p_i^{-1} \tau_i \le \rho^{-1}$ and $g_i(0) = 0$ for all $i \in [k]$. Hence

$$\begin{aligned}
d(x, \vec{0}) &= \left(\sum_{i \in [k]} \mathbf{1}_{p_i \ne 1} p_i^{-1} g_i(x)^4 \right)^{1/2} \\
&\overset{(i)}{\le} \left(\sum_{i \in [k]} \mathbf{1}_{p_i \ne 1} p_i^{-1} \tau_i \cdot x^\top \mathbf{A}^\top \mathbf{W} \mathbf{A} x \cdot g_i(x)^2 \right)^{1/2} \\
&\overset{(ii)}{\le} \rho^{-1/2} (x^\top \mathbf{A}^\top \mathbf{W} \mathbf{A} x)^{1/2} \left(\sum_{i \in [k]} g_i(x)^2 \right)^{1/2} \\
&\overset{(iii)}{\le} \rho^{-1/2} f_{\mathcal{G}}(x) \le \rho^{-1/2}.
\end{aligned}$$

Here, (i) follows from Lemma 1.5, (ii) follows from $\mathbf{1}_{p_i \ne 1} p_i^{-1} \le \rho^{-1} \tau_i^{-1}$ as noted, and (iii) follows from Lemma 1.4. □

Next we will construct nets T_N for $N \ge C \log m$ for sufficiently large constant C that will show show that the contribution of those terms to (9) is negligible. At this scale we have $|T_N| = 2^{2^N} = \exp(\operatorname{poly}(m))$, while there are only m vectors a_i. Consequently our net will simply just approximate each inner product $|\langle a_i, x \rangle|$ up to additive δ accuracy for properly chosen δ. This creates $(1/\delta)^m$ net centers, which is much less than the allowed $2^{2^N} \ge \exp(\operatorname{poly}(m))$.

Lemma 3.7 (Large N). *Consider group leverage score overestimates τ and corresponding weights w (Definition 1.3), $x \in B$, and p_i given by (3). For all $N \ge 0$, there is $T_N \subseteq B$ with $|T_N| \le 2^{2^N}$ and $d(x, T_N) \le 2 \cdot 2^{-2^{N-1}}/m$ for all $x \in B$.*

Proof. Fix an N. Recall from previous arguments (e.g. Lemma 3.6) that

$$\mathbf{1}_{p_i \ne 1} p_i^{-1} g_i(x)^2 \le \mathbf{1}_{p_i \ne 1} p_i^{-1} \tau_i x^\top \mathbf{A}^\top \mathbf{W} \mathbf{A} x \le \rho^{-1} < 1,$$

by Lemmas 1.4 and 1.5. For each $x \in B$ and $\delta = 2^{-2^N}/m$, consider the vector $v^x \in \mathbb{R}^k$ defined as

$$v_i^x \overset{\text{def}}{=} \delta \lfloor \mathbf{1}_{p_i \ne 1} p_i^{-1} g_i(x)^2 / \delta \rfloor.$$

Note that v_i^x can only be one of at most $(1/\delta)^k \le (1/\delta)^m$ distinct vectors. Thus, we can pick T_N to contain one representative $x \in B$ for each distinct v_i^x, and $|T_N| \le 2^{2^N}$ because $(1/\delta)^m = 2^{2^N}$. For any $x \in B$, let y be such that $v^y = v^x$. The result then follows as

$$\begin{aligned}
d(x, T_N) \le d(x, y) &= \left(\sum_{i \in [k]} \mathbf{1}_{\{p_i \ne 1\}} p_i^{-1} (g_i(x)^2 - g_i(y)^2)^2 \right)^{1/2} \\
&\le \left(\sum_{i \in [k]} \mathbf{1}_{\{p_i \ne 1\}} p_i^{-1} |g_i(x)^2 - g_i(y)^2| \cdot (g_i(x)^2 + g_i(y)^2) \right)^{1/2} \\
&\le \left(\sum_{i \in [k]} \delta (g_i(x)^2 + g_i(y)^2) \right)^{1/2} \le \sqrt{\delta (f_{\mathcal{G}}(x) + f_{\mathcal{G}}(y))} \le \sqrt{2\delta} \\
&\le 2 \cdot 2^{-2^{N-1}}/m.
\end{aligned}$$

□

This means that the terms $N \ge 4(\log m + \log_2 \log(1/\varepsilon))$ in (9) have low contribution, as

$$\sum_{N \ge 4(\log m + \log_2 \log(1/\varepsilon))} 2^{N/2} \cdot 2 \cdot 2^{-2^{N-1}}/m \le \varepsilon.$$

We conclude this section by bounding the remaining terms in (9).

Lemma 3.8. *Consider group leverage score overestimates τ and corresponding weights w (Definition 1.3), $x \in B$, and probabilities p_i given by (3). For all $N \ge 0$, there is $T_N \subseteq B$ with $|T_N| \le 2^{2^N}$ and $d(x, T_N) \le C_4 \rho^{-1/2} 2^{-N/2} \sqrt{\log m}$ where C_4 is an absolute constant.*

The proof of this lemma uses the following result, which gives a covering of the unit ball with balls of radius η in the norm $\max_{i \in [m]} |\langle u_i, x \rangle|$ for unit vectors $u_1, \ldots, u_m \in \mathbb{R}^n$.

Theorem 3.9 (Theorem VI.1 of [1]). *Let $u_1, \ldots, u_m \in \mathbb{R}^n$ be vectors with $\|u_i\|_2 \le 1$ for all $i \in [m]$, and $\eta > 0$. There is a universal constant C_3 such that the ball $B_2 \overset{\text{def}}{=} \{x : x \in \mathbb{R}^n, \|x\|_2 \le 1\}$ can be covered with at most $S = m^{C_3/\eta^2}$ subsets $P_1, \ldots, P_S$ satisfying*

$$\max_{\substack{i \in [m], j \in [S] \\ x, y \in P_j}} |\langle u_i, x - y \rangle| \le \eta.$$

Proof of Lemma 3.8. Define $u_j = \tau_i^{-1/2}(\mathbf{A}^\top\mathbf{W}\mathbf{A})^{-1/2}a_j$ for $j \in S_i$. Note that for any $x \in B$ we have $\|(\mathbf{A}^\top\mathbf{W}\mathbf{A})^{1/2}x\|_2 \le 1$ by Lemma 1.4 and $x \in B$. In addition,

$$\langle a_j, x\rangle = \langle(\mathbf{A}^\top\mathbf{W}\mathbf{A})^{-1/2}a_j, (\mathbf{A}^\top\mathbf{W}\mathbf{A})^{1/2}x\rangle = \tau_i^{1/2}\langle u_j, (\mathbf{A}^\top\mathbf{W}\mathbf{A})^{1/2}x\rangle$$

and $\|u_j\|_2 = \tau_i^{-1}a_j^\top(\mathbf{A}^\top\mathbf{W}\mathbf{A})^{-1}a_j \le 1$ by Definition 1.3. Let η satisfy $m^{C_3/\eta^2} = 2^{2^N}$, so $\eta = \sqrt{C_3}2^{-N/2}\sqrt{\log_2 m}$, and let $P_1, \dots, P_S$ be the sets guaranteed by Theorem 3.9 for the vectors u_i and parameter η. Note that the above facts guarantee

$$\max_{\substack{i\in[m], j\in[S]\\ z,w\in P_j}} |\langle u_i, z-w\rangle| \le \eta. \tag{11}$$

For each i let v_i be an arbitrary point from P_i, and let T_N be the set of $(\mathbf{A}^\top\mathbf{W}\mathbf{A})^{-1/2}v_i$ for all i.

For $x \in B$, let P_j be a subset containing $(\mathbf{A}^\top\mathbf{W}\mathbf{A})^{1/2}x$. Such a j must exist since $P_1, \dots P_s$ cover the unit ball and $\|(\mathbf{A}^\top\mathbf{W}\mathbf{A})^{1/2}x\|_2 \le 1$. Let $y = (\mathbf{A}^\top\mathbf{W}\mathbf{A})^{-1/2}v_j$. Then

$$\begin{aligned}
d(x,T_N) \le d(x,y) &= \left(\sum_{i\in[k]} \mathbf{1}_{\{p_i\neq 1\}}p_i^{-1}(g_i(x)^2 - g_i(y)^2)^2\right)^{1/2}\\
&\le \left(\sum_{i\in[k]} \mathbf{1}_{\{p_i\neq 1\}}p_i^{-1}\max_{j\in S_i}(g_i(x)-g_i(y))^2(g_i(x)+g_i(y))^2\right)^{1/2}\\
&\overset{(i)}{\le} \left(\sum_{i\in[k]} \mathbf{1}_{\{p_i\neq 1\}}p_i^{-1}\max_{j\in S_i}\langle a_j, x-y\rangle^2(g_i(x)+g_i(y))^2\right)^{1/2}\\
&\overset{(ii)}{\le} \left(\sum_{i\in[k]} \mathbf{1}_{\{p_i\neq 1\}}p_i^{-1}\tau_i\eta^2(g_i(x)+g_i(y))^2\right)^{1/2}\\
&\overset{(iii)}{\le} \eta\rho^{-1/2}\left(\sum_{i\in[k]} 2(g_i(x)^2+g_i(y)^2)\right)^{1/2}\\
&\le 2\eta\rho^{-1/2} = 2\sqrt{C_3}\rho^{-1/2}2^{-N/2}\sqrt{\log_2 m}.
\end{aligned}$$

Here, (i) follows from $|g_i(x)-g_i(y)| \le g_i(x-y) = \max_{j\in S_i}|\langle a_j, x-y\rangle|$, (ii) holds because $(\mathbf{A}^\top\mathbf{W}\mathbf{A})^{1/2}x, (\mathbf{A}^\top\mathbf{W}\mathbf{A})^{1/2}y \in P_j$ and

$$\langle a_j, x-y\rangle^2 = \tau_i\langle u_j, (\mathbf{A}^\top\mathbf{W}\mathbf{A})^{1/2}x - (\mathbf{A}^\top\mathbf{W}\mathbf{A})^{1/2}y\rangle^2 \le \tau_i\eta^2$$

by Equation (11) and (ii) follows from the choice of p_i. The claim follows from choosing $C_4 = 2\sqrt{C_3}(\log 2)^{-1/2}$. □

With these facts, we now complete the proof of Theorem 3.1.

Proof of Theorem 3.1. Observe that we may assume $\varepsilon \ge 1/m$, as otherwise we can simply return $\mathcal{G}$ as our output sparsifier. Similarly, we may assume $m \ge n$, or else $\mathcal{G}$ itself is a good enough sparsifier. We bound the constant γ in Theorem 3.4 using Lemmas 3.6 to 3.8. For the sets $T_s, T_{s+1}, \dots$ constructed in Lemmas 3.7 and 3.8, we have

$$\gamma \le \sup_{x\in B} 2^{s/2}\cdot d(x,\vec{0}) + \sum_{N\ge s} 2^{N/2}d(x,T_N). \tag{12}$$

For the choices $s = \lceil\log_2\log n\rceil$ and $Z = 2\log(8(1+C_2)m)$, write

$$\sum_{N\ge s} 2^{N/2}d(x,T_N) = \sum_{N\in[s,Z]} 2^{N/2}d(x,T_N) + \sum_{N\ge Z} 2^{N/2}d(x,T_N).$$

Lemma 3.7 implies

$$\begin{aligned}
\sum_{N\ge Z} 2^{N/2}d(x,T_N) &\le 2\sum_{N\ge Z} 2^{N/2-2^{N-1}/m} \le 2\sum_{N\ge Z} 2^{-2^N/(4m)}\\
&\le \frac{1}{100C_2m} \le \frac{\varepsilon}{100C_2}.
\end{aligned}$$

On the other hand, for $\rho = C_1\varepsilon^{-2}\log^3 m$ Lemma 3.8 implies

$$\begin{aligned}
\sum_{N\in[s,Z]} 2^{N/2}d(x,T_N) &\le \sum_{N\in[s,Z]} C_4\rho^{-1/2}\sqrt{\log m}\\
&\le \frac{C_4Z\varepsilon\sqrt{\log m}}{\sqrt{C_1}\log^{3/2} m} = \frac{2C_4\varepsilon\log(8(1+C_2)m)}{\sqrt{C_1}\log m}.
\end{aligned}$$

Finally, Lemma 3.6 implies $d(x,0) \le \rho^{-1/2}$. Plugging these into (12) and using $m \ge n$ yields

$$\gamma \le \frac{2\varepsilon\sqrt{\log m}}{\sqrt{C_1}\log^{3/2} m} + \frac{2C_4\varepsilon\log(8(1+C_2)m)}{\sqrt{C_1}\log m} + \frac{\varepsilon}{100C_2}$$

As $m \ge 2$ without loss of generality we have

$$8(1+C_2)m \le m^{4+\log_2(1+C_2)}:$$

the above yields

$$\gamma \le \frac{2\varepsilon}{\sqrt{C_1}} + \frac{(8+2\log_2(1+C_2))C_4\varepsilon}{\sqrt{C_1}} + \frac{\varepsilon}{100C_2}.$$

For $C_1 = 2C_2^2(2+(8+2\log_2(1+C_2))C_4)^2$ the above gives $\gamma \le \frac{\varepsilon}{C_2}$: the result follows from Theorem 3.4. □

We finally discuss how to iterate the bound Theorem 3.1 to construct hypergraph sparsifiers with $O(n\varepsilon^{-2}\log^3(n/\varepsilon))$ edges. We define $\mathcal{G}_0 = \mathcal{G}$ and $k_0 = k$, and iteratively construct a new hypergraph $\mathcal{G}_{i+1}$ from $\mathcal{G}_i$ with at most $k_{i+1} = 0.9k_i$ hyperedges. To do so, we call Theorem 3.1 on $\mathcal{G}_i$ and ensure the output $\mathcal{G}_{i+1}$ is a $(1+\varepsilon_i)$-spectral sparsifier of $\mathcal{G}_i$ with $\varepsilon_i = \Theta\left(\sqrt{\frac{n\log^3 m}{k_i}}\right)$. For an appropriately-chosen implicit constant the output has at most $0.9k_i$ edges as desired. If this iteration is continued until $\mathcal{G}_r$ has $O(n\varepsilon^{-2}\log^3 m)$ edges, $\mathcal{G}_r$ is a $\prod_{i=0}^{r-1}(1+\varepsilon_i)$-sparsifier of $\mathcal{G}$, where

$$\begin{aligned}
\prod_{i=0}^{r-1}(1+\varepsilon_i) &\le \exp\left(\sum_{i=0}^{r-1}\varepsilon_i\right)\\
&\le \exp\Theta\left(\sqrt{\frac{n\log^3 m}{k_r}}\right)\\
&= \exp\Theta(\varepsilon) = 1+O(\varepsilon),
\end{aligned}$$

implying the result.

4 IMPROVED SIZE BOUND FROM CHAINING

In this section, we obtain an improved size bound of $O(n\varepsilon^{-2}\log m\log r)$ using a more sophisticated chaining argument. Before we begin, it is helpful to describe how we differ from the result obtained in Section 3. Informally, the analysis of the previous section constructed sets T_i where $\sup_{x\in B} d(x,T_i)$ was sufficiently

small. These give a bound on γ, as

$$\gamma \le \sup_{x\in B} 2^{s/2}\cdot d(x,\vec{0}) + \sum_{N\ge s} 2^{N/2} d(x,T_N)$$
$$\le 2^{s/2}\left(\sup_{x\in B} d(x,\vec{0})\right) + \sum_{N\ge s} 2^{N/2}\left(\sup_{x\in B} d(x,T_N)\right).$$

Constructing the sets T_i in turn is relatively straightforward, as a simple greedy packing argument reduces the problem to estimating the *entropy numbers*[1] of B with respect to the distance d.

Unfortunately, the bounds obtained by this technique (first developed in an explicit form by Dudley [12]) are suboptimal in many settings: our approach for bounding $\sup_{x\in B} d(x,T_i)$ is essentially tight (Theorem 3.9 is tight up to constant factors in the exponent, as is stated in Theorem 6.1 of [1]), and the analysis loses from up to $O(\log m)$ levels of the scale parameter N. On the other hand, the expression for γ critically takes the supremum over the *sum* of all scales: if only a small number of the values $d(x,T_i)$ can be near the supremum for a fixed x, the resulting bound we obtain can be significantly tighter.

Actually exploiting this potential for amortization is challenging however, as doing so seems to require additional geometric structure of the metric distance d. In this section we employ a chaining framework of [37] based on *growth functionals*, a powerful technique which uses the geometry of the space of events to control γ. This framework is based on providing a sequence of functions F_i satisfying a certain growth condition. Our approach in this section mirrors previous applications [28, 37] of the framework in proving matrix concentration bounds for sums of rank-1 matrices. However, our setting introduces additional complications beyond the matrix setting, which we briefly discuss here.

A key source of difficulty in applying the technique of [28, 37] is the fact that $f_{\mathcal{G}}(x)$ is not strongly convex. This strongly differs from the rank-1 Chernoff setting, where the sum of rank-1 matrices yields $x^\top\left(\sum_i v_i v_i^\top\right)x$, which *is* strongly convex in the matrix norm formed by $\sum_i v_i v_i^\top$. Strong convexity enables us to prove lower bounds on the difference of growth functionals (**??**): in the rank-1 matrix case this property immediately allows us to obtain the optimal sparsity bounds. Without this property (as noted in [37]), the growth functional framework seems to break down at $\rho = \varepsilon^{-2}\log^2 m$. While we have access to a natural matrix $\mathbf{A}^\top\mathbf{W}\mathbf{A}$ to perform the analysis in, it unfortunately does not approximate $f_{\mathcal{G}}(x)$ well enough for our purposes: for some vectors x we may have $x^\top\mathbf{A}^\top\mathbf{W}\mathbf{A}x \ll f_{\mathcal{G}}(x)$. Thus, we perform our analysis in a "mixed" $(w+f)$-norm which contains both $\mathbf{A}^\top\mathbf{W}\mathbf{A}$ and the energy $f_{\mathcal{G}}(x)$ and establish a strong convexity bound which suffices for our purposes.

A secondary issue related to strong convexity arises from the distance function d defined in the previous section. A feature of the growth functional framework its use of "well-separated" sets (**??**), which have small d-diameter but are in some sense "far apart" under d. However, the analysis of our growth functional requires a stronger property: namely, that convex hulls of the well-separated sets have small d-diameter. If $d(x,\cdot)$ (for every fixed x) were a convex function, this fact would hold immediately: however we believe that convex combinations of points may grow the d-distance arbitrarily. To avoid this issue, we introduce a carefully designed proxy distance function $\widehat{d}$ which overestimates d. We show that while $\widehat{d}(x,\cdot)$ is still not convex (and in fact does not satisfy the triangle inequality) it has these properties in an approximate sense (Lemmas 4.2 and 4.3 in the full version) which suffices for our analysis.

We state the main technical result of this section.

Theorem 4.1. *Let $\mathcal{G} = (\mathcal{S},\mathbf{A})$ be a unit hypergraph, and let τ be given group leverage scores (Definition 1.3) with valid weights w (which do not need to be known). For any constant C, there is an absolute constant C_{13} (which depends on C) such that the matrix hypergraph $\mathcal{H} = \texttt{Subsample}(\mathcal{G},\tau,\rho)$ (Algorithm 1) with $\rho = C_{13}\varepsilon^{-2}\log m\log r$ satisfies with probability at least $1-n^{-C}$ the following:*

$$(1-\varepsilon)f_{\mathcal{G}}(x) \le f_{\mathcal{H}}(x) \le (1+\varepsilon)f_{\mathcal{H}}(x) \text{ for all } x\in\mathbb{R}^n$$

The proof of this theorem is deferred to the full version in https://arxiv.org/pdf/2209.10539v1.pdf.

ACKNOWLEDGMENTS

We thank James Lee for coordinating submissions.

Yang P. Liu is supported by the Google PhD Fellowship Program. Aaron Sidford is supported by a Microsoft Research Faculty Fellowship, NSF CAREER Award CCF-1844855, NSF Grant CCF-1955039, a PayPal research award, and a Sloan Research Fellowship.

REFERENCES

[1] Noga Alon and Bo'az Klartag. 2017. Optimal compression of approximate inner products and dimension reduction. In *58th Annual IEEE Symposium on Foundations of Computer Science—FOCS 2017*. IEEE Computer Soc., Los Alamitos, CA, 639–650. https://doi.org/10.1109/FOCS.2017.65

[2] Nikhil Bansal, Ola Svensson, and Luca Trevisan. 2019. New Notions and Constructions of Sparsification for Graphs and Hypergraphs. In *FOCS*. IEEE Computer Society, 910–928. https://doi.org/10.1109/FOCS.2019.00059

[3] Joshua D. Batson, Daniel A. Spielman, and Nikhil Srivastava. 2014. Twice-Ramanujan Sparsifiers. *SIAM Rev.* 56, 2 (2014), 315–334. https://doi.org/10.1137/090772873

[4] András A. Benczúr and David R. Karger. 2015. Randomized Approximation Schemes for Cuts and Flows in Capacitated Graphs. *SIAM J. Comput.* 44, 2 (2015), 290–319. https://doi.org/10.1137/070705970

[5] Jean Bourgain, Joram Lindenstrauss, and Vitali Milman. 1989. Approximation of zonoids by zonotopes. *Acta mathematica* 162 (1989), 73–141.

[6] Chandra Chekuri and Chao Xu. 2018. Minimum cuts and sparsification in hypergraphs. *SIAM J. Comput.* 47, 6 (2018), 2118–2156. https://doi.org/10.1137/18M1163865

[7] Yu Chen, Sanjeev Khanna, and Ansh Nagda. 2020. Near-linear Size Hypergraph Cut Sparsifiers. In *FOCS*. IEEE, 61–72. https://doi.org/10.1109/FOCS46700.2020.00015

[8] Michael B. Cohen, Ben Cousins, Yin Tat Lee, and Xin Yang. 2019. A near-optimal algorithm for approximating the John Ellipsoid. In *COLT (Proceedings of Machine Learning Research, Vol. 99)*. PMLR, 849–873.

[9] Michael B. Cohen, Rasmus Kyng, Gary L. Miller, Jakub W. Pachocki, Richard Peng, Anup Rao, and Shen Chen Xu. 2014. Solving SDD linear systems in nearly $m\log^{1/2} n$ time. In *STOC*. 343–352. https://doi.org/10.1145/2591796.2591833

[10] Michael B. Cohen, Yin Tat Lee, Cameron Musco, Christopher Musco, Richard Peng, and Aaron Sidford. 2015. Uniform Sampling for Matrix Approximation. In *ITCS*. ACM, 181–190. https://doi.org/10.1145/2688073.2688113

[11] Michael B. Cohen and Richard Peng. 2015. L_p Row Sampling by Lewis Weights. In *STOC*. ACM, 183–192. https://doi.org/10.1145/2746539.2746567

[12] R. M. Dudley. 1967. The sizes of compact subsets of Hilbert space and continuity of Gaussian processes. *J. Functional Analysis* 1 (1967), 290–330. https://doi.org/10.1016/0022-1236(67)90017-1

[1] The n^{th} entropy number of a set B with respect to distance d is the smallest ϵ such that there exists a set T with $|T| \le 2^{2^n}$ and $d(x,T) \le \varepsilon$ for all $x \in B$. Lemma 3.8 in fact bounds exactly these entropy numbers, although we do not call them that explicitly.

[13] X. Fernique. 1975. Regularité des trajectoires des fonctions aléatoires gaussiennes. In *École d'Été de Probabilités de Saint-Flour, IV-1974*. Springer, Berlin, 1–96.

[14] Arun Jambulapati, Yang P. Liu, and Aaron Sidford. 2022. Improved iteration complexities for overconstrained p-norm regression. In *STOC*. ACM, 529–542. https://doi.org/10.1145/3519935.3519971

[15] Arun Jambulapati and Aaron Sidford. 2021. Ultrasparse ultrasparsifiers and faster laplacian system solvers. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM, 540–559. https://doi.org/10.5555/3458064.3458097

[16] Michael Kapralov, Robert Krauthgamer, Jakab Tardos, and Yuichi Yoshida. 2021. Spectral Hypergraph Sparsifiers of Nearly Linear Size. In *FOCS*. IEEE, 1159–1170. https://doi.org/10.1109/FOCS52979.2021.00114

[17] Michael Kapralov, Robert Krauthgamer, Jakab Tardos, and Yuichi Yoshida. 2021. Towards tight bounds for spectral sparsification of hypergraphs. In *STOC*. ACM, 598–611. https://doi.org/10.1145/3406325.3451061

[18] Jonathan A. Kelner, Lorenzo Orecchia, Aaron Sidford, and Zeyuan Allen Zhu. 2013. A simple, combinatorial algorithm for solving SDD systems in nearly-linear time. In *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*. 911–920. https://doi.org/10.1145/2488608.2488724

[19] Dmitry Kogan and Robert Krauthgamer. 2015. Sketching cuts in graphs and hypergraphs. In *Proceedings of the 2015 Conference on Innovations in Theoretical Computer Science*. 367–376.

[20] Ioannis Koutis, Gary L. Miller, and Richard Peng. 2010. Approaching Optimality for Solving SDD Linear Systems. In *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010, October 23-26, 2010, Las Vegas, Nevada, USA*. 235–244. https://doi.org/10.1137/110845914

[21] Ioannis Koutis, Gary L. Miller, and Richard Peng. 2011. A Nearly-$m \log n$ Time Solver for SDD Linear Systems. In *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011, Palm Springs, CA, USA, October 22-25, 2011*. 590–598. https://doi.org/10.1109/FOCS.2011.85

[22] Rasmus Kyng, Yin Tat Lee, Richard Peng, Sushant Sachdeva, and Daniel A. Spielman. 2016. Sparsified Cholesky and multigrid solvers for connection laplacians. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, Daniel Wichs and Yishay Mansour (Eds.). ACM, 842–850. https://doi.org/10.1145/2897518.2897640

[23] Rasmus Kyng and Sushant Sachdeva. 2016. Approximate Gaussian Elimination for Laplacians - Fast, Sparse, and Simple. In *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*, Irit Dinur (Ed.). IEEE Computer Society, 573–582. https://doi.org/10.1109/FOCS.2016.68

[24] James R Lee. 2022. Spectral hypergraph sparsification via chaining. *arXiv preprint arXiv:2209.04539* (2022).

[25] Mu Li, Gary L Miller, and Richard Peng. 2013. Iterative row sampling. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*. IEEE, 127–136. https://doi.org/10.1109/FOCS.2013.22

[26] Cameron Musco, Christopher Musco, David P. Woodruff, and Taisuke Yasuda. 2022. Active Linear Regression for ℓ_p Norms and Beyond. In *63rd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2022, Denver, CO, USA, October 31 - November 3, 2022*. IEEE, 744–753. https://doi.org/10.1109/FOCS54457.2022.00076

[27] Richard Peng and Daniel A. Spielman. 2014. An efficient parallel solver for SDD linear systems. In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, David B. Shmoys (Ed.). ACM, 333–342. https://doi.org/10.1145/2591796.2591832

[28] Mark Rudelson. 1996. Random vectors in the isotropic position. *MSRI Preprint 1996-060* (1996).

[29] Gideon Schechtman. 2011. Tight embedding of subspaces of L_p into ℓ_p^N for even p. *Proc. Amer. Math. Soc.* 139, 12 (2011), 4419–4421.

[30] Gideon Schechtman and Artem Zvavitch. 2001. Embedding Subspaces of L_p into ℓ_p^N, $0 < p < 1$. *Mathematische Nachrichten* 227, 1 (2001), 133–142.

[31] Tasuku Soma and Yuichi Yoshida. 2019. Spectral Sparsification of Hypergraphs. In *SODA*. SIAM, 2570–2581. https://doi.org/10.1137/1.9781611975482.159

[32] D. Spielman and S. Teng. 2014. Nearly Linear Time Algorithms for Preconditioning and Solving Symmetric, Diagonally Dominant Linear Systems. *SIAM J. Matrix Anal. Appl.* 35, 3 (2014), 835–885. https://doi.org/10.1137/090771430 Available at http://arxiv.org/abs/cs/0607105.

[33] Daniel A. Spielman and Nikhil Srivastava. 2011. Graph sparsification by effective resistances. *SIAM J. Comput.* 40, 6 (2011), 1913–1926. https://doi.org/10.1137/080734029

[34] Michel Talagrand. 1987. Regularity of Gaussian processes. *Acta Math.* 159, 1-2 (1987), 99–149. https://doi.org/10.1007/BF02392556

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

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

[37] Michel Talagrand. 2014. *Upper and lower bounds for stochastic processes*. Vol. 60. Springer.

[38] Prasad Tetali. 1991. Random walks and the effective resistance of networks. *J. Theoret. Probab.* 4, 1 (1991), 101–109. https://doi.org/10.1007/BF01046996

Received 2022-11-07; accepted 2023-02-06