Spectral Hypergraph Sparsifiers of Nearly Linear Size

Michael Kapralov[*]
École Polytechnique Fédérale de Lausanne
michael.kapralov@epfl.ch

Robert Krauthgamer[†]
Weizmann Institute of Science
robert.krauthgamer@weizmann.ac.il

Jakab Tardos[‡]
École Polytechnique Fédérale de Lausanne
jakab.tardos@epfl.ch

Yuichi Yoshida[§]
National Institute of Informatics
yyoshida@nii.ac.jp

June 3, 2021

Abstract

Graph sparsification has been studied extensively over the past two decades, culminating in spectral sparsifiers of optimal size (up to constant factors). Spectral hypergraph sparsification is a natural analogue of this problem, for which optimal bounds on the sparsifier size are not known, mainly because the hypergraph Laplacian is non-linear, and thus lacks the linear-algebraic structure and tools that have been so effective for graphs.

Our main contribution is the first algorithm for constructing ϵ-spectral sparsifiers for hypergraphs with $O^*(n)$ hyperedges, where O^* suppresses $(\epsilon^{-1} \log n)^{O(1)}$ factors. This bound is independent of the rank r (maximum cardinality of a hyperedge), and is essentially best possible due to a recent bit complexity lower bound of $\Omega(nr)$ for hypergraph sparsification.

This result is obtained by introducing two new tools. First, we give a new proof of spectral concentration bounds for sparsifiers of graphs; it avoids linear-algebraic methods, replacing e.g. the usual application of the matrix Bernstein inequality and therefore applies to the (non-linear) hypergraph setting. To achieve the result, we design a new sequence of hypergraph-dependent ϵ-nets on the unit sphere in $\mathbb{R}^n$. Second, we extend the weight-assignment technique of Chen, Khanna and Nagda [FOCS'20] to the spectral sparsification setting. Surprisingly, the number of spanning trees after the weight assignment can serve as a potential function guiding the reweighting process in the spectral setting.

Contents

1 Introduction

We study spectral sparsification of hypergraphs, where the goal is to reduce the size of a hypergraph while preserving its energy. Given a hypergraph $H = (V, E, w)$ with a weight function $w : E \to \mathbb{R}_+$ over its hyperedges, the *energy* of $x \in \mathbb{R}^V$ (called a potential vector) is defined as

$$Q_H(x) := \sum_{e \in E} w(e) \cdot \max_{u,v \in e} (x_u - x_v)^2.$$

The problem of minimizing $Q_H(x)$ over $x \in \mathbb{R}^V$ subject to certain constraints appears in many problems involving hypergraphs, including clustering [TMIY20], semi-supervised learning [HSJR13, YNY$^+$19, ZHTC20] and link prediction [YNN$^+$20], from which we can see the relevance of $Q_H(x)$ in application domains. Note that when $x \in \mathbb{R}^V$ is a characteristic vector $\mathbb{1}_S \in \{0,1\}^V$ of a vertex subset $S \subset V$, the energy $Q_H(\mathbb{1}_S)$ coincides with the total weight of hyperedges cut by S, where we say that a hyperedge $e \in E$ is *cut* by S if $e \cap S \neq \emptyset$ and $e \cap (V \setminus S) \neq \emptyset$.

Since the number of hyperedges in a hypergraph of n vertices can be $\Omega(2^n)$, it is desirable to reduce the number of hyperedges in the hypergraph while (approximately) preserving the value of $Q_H(x)$ for every $x \in \mathbb{R}^V$, because this lets us speed up any algorithm involving Q_H and reduce its memory usage by running it on the smaller hypergraph instead of H itself. Soma and Yoshida [SY19] formalized this concept as spectral sparsification for hypergraphs – a natural generalization of the corresponding concept introduced by the celebrated work of [ST11] for graphs. Specifically, for $0 < \epsilon < 1$, we say that a hypergraph $\widetilde{H}$ is an *ϵ-spectral-sparsifier* of a hypergraph H if $\widetilde{H}$ is a reweighted subgraph of H such that

$$\forall x \in \mathbb{R}^V, \qquad Q_{\widetilde{H}}(x) \in (1 \pm \epsilon) Q_H(x).^1 \tag{1}$$

We note that when H is an ordinary graph, this definition matches that for graphs [ST11]. Soma and Yoshida [SY19] showed that every hypergraph H admits an ϵ-spectral-sparsifier with $\widetilde{O}(n^3/\epsilon^2)$ hyperedges,[2] and gave a polynomial-time algorithm for constructing such sparsifiers. Since then the number of hyperedges needed has been reduced to $\widetilde{O}(nr^3/\epsilon^2)$ [BST19], and recently to $\widetilde{O}(nr/\epsilon^{O(1)})$ [KKTY21], where r is the maximum size of a hyperedge in the input hypergraph H (called the *rank* of H).

The natural question whether every hypergraph admits a spectral sparsifier with $\widetilde{O}(n)$ hyperedges (for fixed ϵ) has proved to be challenging. On the one hand, it is well-known that a hypergraph is a strictly richer object than an ordinary graph (hyperedges cannot be "simulated" by edges, even approximately), and in all previous results and techniques, this extra complication introduced an extra factor of at least r. On the other hand, an exciting recent result [CKN20] has achieved sparsifiers with $\widetilde{O}(n)$ hyperedges, if one is only interested in preserving the *hypergraph cut function*, i.e., satisfying (1) only for all characteristic vectors $x = \mathbb{1}_S$ where $S \subseteq V$. Nevertheless, the spectral version of this question has remained open, primarily due to the non-linearity of the hypergraph Laplacian and the lack of linear-algebraic tools that have been so effective for graphs.

We settle this question by showing that a nearly linear number of hyperedges suffices.

Theorem 1. *For every hypergraph with n vertices and every $1/n \leq \epsilon \leq 1/2$, there exists an ϵ-spectral-sparsifier with $O(n\epsilon^{-4} \log^3 n)$ hyperedges. Moreover, one can construct such a sparsifier in time $\widetilde{O}(mr + \operatorname{poly}(n))$, where m is the number of hyperedges and r is the maximum size of a hyperedge in H.*

We note that the bit complexity of our sparsifier is tight up to a polylogarithmic factor for a large range of r due to the lower bound of [KKTY21].

---

[1] $a \in (1 \pm \epsilon)b$ is a shorthand for $(1 - \epsilon)b \leq a \leq (1 + \epsilon)b$.
[2] Throughout, $\widetilde{O}(\cdot)$ suppresses a factor of $\log^{O(1)} n$.

1.1 Additional Related Work

Recall that we call $\widetilde{H} = (V, \widetilde{E}, \widetilde{w})$ an ϵ-*cut sparsifier* of $H = (V, E, w)$ if every cut weight is preserved to within a factor of $1 \pm \epsilon$. This definition matches the one for ordinary graphs introduced by Benczúr and Karger [BK15], who showed that every graph has an ϵ-cut-sparsifier with $O(n \log n / \epsilon^2)$ edges, where n is the number of vertices. For hypergraphs, Kogan and Krauthgamer [KK15] gave the first construction of non-trivial cut sparsifiers, which uses $O(n(r + \log n)/\epsilon^2)$ hyperedges, where r is the maximum size of a hyperedge. They also mentioned that the results of Newman and Rabinovich [NR13] implicitly give an ϵ-cut sparsifier with $O(n^2/\epsilon^2)$ hyperedges. Chen, Khanna, and Nagda [CKN20] improved this bound to $O(n \log n / \epsilon^2)$, which is almost tight because one needs $\Omega(n/\epsilon^2)$ edges even for ordinary graphs [ACK+16, CKST19].

Spielman and Teng [ST11] introduced the notion of a spectral sparsifier for ordinary graphs and showed that every graph on n vertices admits an ϵ-spectral sparsifier with $O(n \log^{O(1)} n / \epsilon^2)$ edges. This bound was later improved to $O(n/\epsilon^2)$ [BSS12], which is tight [ACK+16, CKST19]. The literature on graph sparsification is too vast to cover here, including [ST11, SS11, BSS12, ZLO15, LS15, LS17] and many other constructions, and we refer the reader to the surveys [Vis13, Ten16].

For an ordinary graph $G = (V, E, w)$, the *Laplacian* of G is the matrix $L_G = D_G - A_G$, where $D_G \in \mathbb{R}^{V \times V}$ is the diagonal (weighted) degree matrix and $A_G \in \mathbb{R}^{V \times V}$ is the adjacency matrix of G. Then, the energy Q_G, defined in (1), can be written also as

$$Q_G(x) = x^\top L_G x.$$

For a hypergraph $H = (V, E, w)$, it is known that we can define a (multi-valued) Laplacian operator $L_H : \mathbb{R}^V \to 2^{\mathbb{R}^V}$, so that

$$Q_H(x) = x^\top y$$

for every $x \in \mathbb{R}^V$ and $y \in L_H(x)$ [Lou15, CLTZ18, Yos19] (hence we can write $Q_H(x)$ also as $x^\top L_H(x)$ without ambiguity). Although the Laplacian operator L_H is no longer a linear operator, its mathematical property has been actively investigated [IMTY19, FSY18, IKT21] through the theory of monotone operators and evolution equations [Kom67, Miy92].

Yoshida [Yos16] proposed a Laplacian operator for directed graphs and used it to study structures of real-world networks. The Laplacian operators for graphs, hypergraphs, and directed graphs mentioned above were later unified and generalized as Laplacian operator for submodular transformations/submodular hypergraphs [LM18, Yos19].

2 Preliminaries

In this paper, we deal with spectral sparsification of hypergraphs. For the sake of generality, we consider weighted hypergraphs denoted $H = (V, E, w)$, where V is the vertex set of size n, E is the hyperedge set of size m, and $w : E \to \mathbb{R}_+$ is the set of hyperedge weights. We will also, however, deal with ordinary graphs, that is graphs where each edge contains two vertices exactly. In order to distinguish clearly between graphs and hypergraphs, we will typically denote graphs as $G = (V, F, z)$, where V is the vertex set, F is the edge set, and $z : F \to \mathbb{R}_+$ is the set of edge weights. In general we will use f and g to denote ordinary edges, while reserving e to denote hyperedges.

For simplicity *all graphs and hypergraphs we consider in this paper will be connected.*

2.1 Spectral Graph Theory

Definition 2.1. *The Laplacian of a weighted graph $G = (V, F, z)$ is defined as the matrix $L_G \in \mathbb{R}^{V \times V}$ such that*

$$(L_G)_{uv} = \begin{cases} d(u) & \text{if } u = v, \\ -z(u,v) & \text{if } (u,v) \in F, \\ 0 & \text{otherwise.} \end{cases}$$

Here $d(u)$ denotes the weighted degree of u, that is the sum of all weights of incident edges. Thus L_G is a positive semidefinite matrix, and its quadratic form can be written as

$$x^\top L_G x = \sum_{(u,v) \in F} z(u,v) \cdot (x_u - x_v)^2.$$

The spectral sparsifier of G is defined as a reweighted subgraph which closely approximates the quadratic form of the Laplacian on every possible vector.

Definition 2.2. *Let $G = (V, F, z)$ be a weighted ordinary graph. Let $\widetilde{G} = (V, \widetilde{F}, \widetilde{z})$ be a reweighted subgraph of G, defined by $\widetilde{z} : F \to \mathbb{R}_+$, where $\widetilde{F} = \{f \in F \mid \widetilde{z}(f) > 0\}$. For $\epsilon > 0$, $\widetilde{G}$ is an ϵ-spectral sparsifier of G if for every $x \in \mathbb{R}^V$*

$$x^\top L_{\widetilde{G}} x \in (1 \pm \epsilon) \cdot x^\top L_G x.$$

The quadratic form of the graph Laplacian from Definition 2.1 can be generalized to hypergraphs. Although this generalization is highly non-linear, we still refer to it as the "quadratic form" of the hypergraph.

Definition 2.3. *The quadratic form (or sometimes energy) of a hypergraph $H = (V, E, w)$ is defined on the input vector $x \in \mathbb{R}^V$ as*

$$Q_H(x) = \sum_{e \in E} w(e) \cdot \max_{u,v \in e}(x_u - x_v)^2.$$

Consequently, we may also define the concept of spectral sparsification in hypergraphs, analogously to Definition 2.2:

Definition 2.4. *Let $H = (V, E, w)$ be a weighted hypergraph. Let $\widetilde{H} = (V, \widetilde{E}, \widetilde{w})$ be a reweighted subgraph of H, defined by $\widetilde{w} : E \to \mathbb{R}_+$, where $\widetilde{E} = \{e \in E \mid \widetilde{w}(e) > 0\}$. For $\epsilon > 0$, $\widetilde{H}$ is an ϵ-spectral sparsifier of H if for every $x \in \mathbb{R}^V$*

$$Q_{\widetilde{H}}(x) \in (1 \pm \epsilon) \cdot Q_H(x).$$

2.2 Effective Resistance

Definition 2.5. *Let $G = (V, F, z)$ be a weighted ordinary graph. The effective resistance of a pair of vertices (u, v) is defined as*

$$R_G(u,v) = (\chi_u - \chi_v)^\top L_G^+ (\chi_u - \chi_v).$$

Here $\chi_u \in \mathbb{R}^V$ is the vector with all zeros, and a single 1 at the coordinate corresponding to u. L_G^+ is the Moore-Penrose pseudo-inverse of L_G, which is positive semidefinite.

We may write $R(u,v)$ in cases where G is clear from context.

We will often use the notation $R_G(f) = R_G(u, v)$ where $f = (u, v)$ is an edge. It is important to note, however, that effective resistance is a function of the vertex pair, not the edge, and does not depend directly on the weight of f.

We now state several well-known and useful facts about effective resistance.

Fact 2.6. *The effective resistance of an edge (u, v) is alternatively defined as*

$$R_G(u, v) = \max_{x \in \mathbb{R}^V} \frac{(x_u - x_v)^2}{x^\top L x}.$$

Fact 2.7. *Effective resistance constitutes a metric on V.*

Fact 2.8. *For any weighted graph $G = (V, F, z)$ and any edge $f \in F$ we have $z(f) \cdot R_G(f) \le 1$, with equality if and only if f is a bridge.*

Fact 2.9. *For any weighted graph $G = (V, F, z)$ we have*

$$\sum_{f \in F} z(f) \cdot R_G(f) = n - 1.$$

2.3 Chernoff Bound

Theorem 2.10 (Chernoff bound, see for example [AS08])**.** *Let $Z_1, Z_2, \ldots, Z_k$ be independent random variables in the range $[0, a]$. Furthermore, let $\sum Z_i = Z$ and let $\mu \ge \mathbb{E}(Z)$. Then for any $\delta \in (0, 1)$,*

$$\mathbb{P}\left(|Z - \mathbb{E}(Z)| \ge \delta \mu\right) \le 2 \exp\left(-\frac{\delta^2 \mu}{3a}\right).$$

3 Technical Overview

3.1 Analyzing Ordinary Graphs

The sparsification of ordinary graphs is a highly studied topic, with several techniques proposed for the construction of spectral sparsifiers throughout the years [ST11, SS11, BSS12, ZLO15, LS15, LS17]. However, the analysis of spectral sparsifiers always relies heavily on the linear nature of the graph Laplacian, e.g., using matrix concentration results such as matrix Bernstein [Tro11] or the work of [RV07]. This presents a significant problem when attempting to generalize these techniques to the highly non-linear setting of hypergraph spectral sparsification. Indeed, all previous results lose at least a factor of r due to this obstacle. We therefore dedicate the entirety of our first technical section (Section 4) to presenting a new proof of the existence of nearly linear spectral sparsifiers for ordinary graphs. We use the algorithm from [SS11], which constructs a sparsifier $\widetilde{G}$ by sampling each edge with probability proportional to its effective resistance. However, our proof avoids using matrix concentration inequalities, and instead relies on a more direct chaining technique for proving the concentration of $x^\top L_{\widetilde{G}} x$ around its expectation, i.e. $x^\top L_G x$, for all x simultaneously. To our knowledge, this is the first nearly-optimal direct analysis of spectral sparsification through effective resistance sampling. It will also be the basis of our main result, as we adapt it to the hypergraph setting in Sections 5 and 6.

More formally, for an input graph $G = (V, F, z)$, we define $\widetilde{G}$ as the result of sampling each edge f of G independently with probability $p(f) \approx z(f) \cdot R_G(f)$, and setting its weight to $\widetilde{z}(f) = z(f)/p(f)$. Our aim is then to prove

$$x^\top L_{\widetilde{G}} x \approx x^\top L_G x \tag{2}$$

simultaneously for all $x \in \mathbb{R}^V$. For simplicity we assume that $x^\top L_G x = 1$. Equation (2) is in fact the concentration of a random variable around its expectation, and so we can use Chernoff bound to prove it for any specific x. Our plan is then to use a combination of Chernoff and union bounds to prove it for all possible x. Since x can take any value in $\mathbb{R}^V$ we must discretize it to some ϵ-net while retaining a good approximation to its quadratic form, i.e. $x^\top L_G x$.

Let us take a closer look at the application of Chernoff bound to Equation (2): $x^\top L_{\widetilde{G}} x$ is the sum of the independent random variables $\widetilde{z}(u, v) \cdot (x_u - x_v)^2$ for $(u, v) \in F$; hence, by Theorem 2.10, the strength of the bound depends crucially on the upper bound a on values that each random individual random variable can possibly attain. The maximum value of $\widetilde{z}(u, v) \cdot (x_u - x_v)^2$ is attained when (u, v) is sampled in $\widetilde{G}$, in which case it is $\approx (x_u - x_v)^2 / R_G(u, v)$. Thus

$$\mathbb{P}\left(x^\top L_{\widetilde{G}} x \not\approx x^\top L_G x\right) \lesssim \exp\left(-\frac{1}{\max_{(u,v)\in F}(x_u - x_v)^2 / R_G(u, v)}\right).$$

This upper bound can be as bad as $\exp(-\widetilde{O}(1))$ and is far too crude for our purposes—no sufficiently sparse rounding scheme (i.e., discretization) exists for x. We turn to the technique of chaining—the use of progressively finer and finer rounding schemes.

As seen above, the strength of our Chernoff bound depends primarily on the quantity $(x_u - x_v)^2 / R_G(u, v)$ for each edge (u, v), which we call the "power" of the edge. Therefore, it makes sense to partition the edges of G into a logarithmic number of classes based on their power, that is F_i contains edges (u, v) for which $(x_u - x_v)^2 \approx 2^{-i} \cdot R_G(u, v)$. When focusing only on the subgraphs $G(F_i)$ induced by F_i, we get the more fine-tuned Chernoff bound

$$\mathbb{P}\left(x^\top L_{\widetilde{G}(F_i)} x \not\approx x^\top L_{G(F_i)} x\right) \lesssim \exp\left(-\frac{1}{\max_{(u,v)\in F_i}(x_u - x_v)^2 / R_G(u, v)}\right) \lesssim \exp\left(-2^i\right).$$

We thus have the task of proposing a rounding scheme $\varphi_i : \mathbb{R}^V \to \mathbb{R}^V$ specially for each class F_i such that

- the image of φ_i is a finite set of size at most $\approx \exp\left(2^i\right)$,

- the rounding approximately preserves the quantity $(x_u - x_v)^2$ for $(u, v) \in F_i$.

To gain more intuition on what such a rounding scheme must look like, we draw inspiration from the idea of resistive embedding from [SS11]. We map the edges in F_i, as well as our potential vector x, into vectors in $\mathbb{R}^n$ in such a way that all the relevant quantities arise as norms or scalar products:

$$(u, v) \mapsto \underline{a_{u,v}} = \frac{L_G^{+/2}(\chi_u - \chi_v)}{\left\| L_G^{+/2}(\chi_u - \chi_v) \right\|},$$

$$x \mapsto \underline{y_x} = L_G^{1/2} x.$$

Notice that both $\underline{a_{u,v}}$ and $\underline{y_x}$ are normalized (since $x^\top L_G x = 1$). Furthermore, the crucial quantity, the power of the edge (u, v) arises as the square of a scalar product:

$$\langle \underline{a_{u,v}}, \underline{y_x} \rangle^2 = \frac{(x^\top(\chi_u - \chi_v))^2}{(\chi_u - \chi_v)^\top L_G^+(\chi_u - \chi_v)} = \frac{(x_u - x_v)^2}{R_G(u, v)}.$$

Thus we are interested in rounding $\underline{y_x}$ in a way that preserves $\langle \underline{a_{u,v}}, \underline{y_x} \rangle^2$ up to small multiplicative error *in all cases where it was $\approx 2^{-i}$ to begin with*. Thus, it suffices to guarantee an additive

error of at most $\lesssim 2^{-i}$ in our rounding scheme. This is the known problem of "compression of approximate inner products" and has been previously studied; [AK17] guarantees a rounding scheme whose image is of size at most $\approx \exp(2^i)$. This can be translated into a rounding scheme for $x \in \mathbb{R}^V$, with the same image-size, exactly as desired (see Lemma 4.2).

With the desired rounding scheme in hand, we can now use a combination of Chernoff and union bounds to prove that for all x simultaneously

$$x^\top L_{\widetilde{G}(F_i)} x \approx x^\top L_{G(F_i)} x.$$

Summing this over all edge-classes gives us Equation 2.

For the detailed proof, which is considerably more complicated than the above sketch, see the proof of Theorem 4.1 in Section 4.

3.2 Extension to Hypergraphs

To adapt the previous argument to the hypergraph setting, we use the idea of balanced weight assignments from [CKN20]. Essentially, we construct an ordinary graph $G = (V, F, z)$ to accompany our input hypergraph $H = (V, E, w)$ by replacing each hyperedge e with a clique F_e over the vertices in e. However, unlike in some previous works on hypergraph sparsification, the clique F_e is not assigned weights uniformly, but instead the weight is carefully distributed among the edges. Intuitively, all the weight is shifted onto the most "important" edges. In the case of [CKN20], the measure of importance was "strength", a quantity relevant to cut sparsification, while in our case it is effective resistance.

More formally, a weighting assignment z of the cliques is considered γ-balanced if for all $e \in E$

- $\sum_{f \in F_e} z(f) = w(e)$,

- and

$$\gamma \cdot \min_{g \in F_e:\ z(g)>0} R_G(g) \geq \max_{f \in F_e} R_G(f).$$

In words, all but the zero-weight edges of F_e have approximately the same effective resistance. This allows hyperedge e to inherit this effective-resistance value as its importance when sampling hyperedges. Our task is now to prove the existence of balanced weight assignments for all hypergraphs, and then to adapt the proof of Section 4.

Finding balanced weight assignments. In [CKN20], balanced weight assignments are constructed through the following intuitive process: Find a pair of edges violating the second constraint, that is $f, g \in F_e$ where $z(g) \neq 0$ and f has significantly higher importance than g. Then shift weight from g to f; this alleviates the constraint violation either because the importances of g and f become more similar, or simply because the weight of g decreases to 0. We call this resolving the imbalance of f and g. [CKN20] strings together such steps, carefully ordered and discretized, to eventually produce a balanced weight assignment of the input hypergraph.

However, their analysis relies heavily on a certain lemma about how "strength" (their measure of edge importance) behaves under weight updates. Lemma 6 of [CKN20] states that altering the weight of an edge f, will not affect edges of significantly greater "strength" than f. This is not the case for effective resistances. It is easy to construct scenarios to the contrary; even ones in which altering the weight of edges of low resistance can increase the maximum effective resistance in the graph.

Thus the analysis of [CKN20] does not extend to our setting. Instead we use a potential function argument to say that we make irreversible progress whenever we resolve the imbalance of two edges f and g. Our choice of potential function is surprising, and is one of the main technical contributions of this paper. We define the spanning tree potential (or ST-potential) of a connected weighted ordinary graph $G = (V, F, z)$, denoted $\Psi(G)$. For edge weights that equal 1 uniformly (that is for unweighted graphs) it is simply the logarithm of the number of distinct spanning trees in G. In weighted graphs it is generalized to

$$\Psi(G) = \log \left(\sum_{T \in \mathbb{T}} \prod_{f \in T} z(f) \right),$$

where $\mathbb{T}$ denotes the set of all spanning trees in G. Due to the relationship between spanning tree sampling and effective resistances (see for example [Lov93]) we can prove a crucial update formula for $\Psi(G)$: if an edge f has its weight changed by $\lambda \in \mathbb{R}$, the ST-potential increases by $\log(1 + \lambda \cdot R(f))$. Since whenever we resolve the imbalance of a pair of edges, we shift weight from the edge of lower effective resistance to that of higher effective resistance, this allows us to argue that the ST-potential always increases throughout the process, which eventually terminates in a balanced weight assignment (see Algorithm 1 and Theorem 5.8).

This proves the existence of balanced weight assignments, which suffices to show the existence of nearly linear size spectral sparsifiers for all hypergraphs. However, to improve running time (from exponential to polynomial in the input size), we introduce the novel concept of *approximate* balanced weight assignments, by slightly relaxing the definition. These are still sufficient to aid in constructing spectral sparsifiers, and are faster to construct using Algorithm 2.

For more details on the ST-potential, as well as the construction of balanced weight assignments see Section 5.

Using balanced weight assignments to construct hypergraph spectral sparsifiers. Given a hypergraph $H = (V, E, w)$ and its balanced weight assignment $G = (V, F, z)$ we assign importance to each hyperedge proportionally to the maximum effective resistance in F_e (the clique corresponding to e). Thus we perform importance sampling, which samples each hyperedge independently with probability $p(e) \approx w(e) \cdot \max_{f \in F_e} R_G(f)$.

The broad strokes of the hypergraph proof in Section 6 proceed very similarly to those of the proof for ordinary graphs in Section 4. However, numerous details need to be figured out in order to bridge the gap between the two settings. It is interesting to note that our rounding scheme is exactly the same as in Section 4, to the point of even being defined in terms of G, not H. (Indeed it is impossible to define such a rounding scheme directly in terms of H; Lemma 4.2 relies heavily on the linear nature of the ordinary graph Laplacian.) Nevertheless, we manage to extend the approximation guarantee of the rounding scheme from edges to hyperedges (see Claim 6.3).

For the detailed analysis of hypergraph spectral sparsification through effective resistance-based importance sampling, see Section 6.

3.3 Speed-Up

Using a results of Sections 5 and 6 we can put together a polynomial time algorithm for spectral sparsification of hypergraphs. Simply run Algorithm 2 to produce an approximate balanced weight assignment, and then use importance sampling (Algorithm 3). The bottleneck of this procedure is constructing the weight assignment, which takes time $O(m \cdot \text{poly}(n))$. (Given the weight assignment, it is trivial to implement importance sampling).

In Section 7 we reduce this to the nearly optimal $\widetilde{O}(mr + \mathrm{poly}\,(n))$. (Note that $O(mr)$ is the size of the input.) Our first step is the common trick of using a faster sparsification algorithm, but one which produces a larger output, to preprocess the input hypergraph. We use the algorithm of [BST19] which – with small modifications – can be made to run in the desired $\widetilde{O}(mr + \mathrm{poly}\,(n))$ time. The resulting hypergraph has only polynomially many hyperedges (in n); however, the aspect ratio of edge weights (that is the ratio between the largest and smallest edge weights) can naturally be exponential in n.

Unfortunately, Algorithm 2 scales linearly in the aspect ratio of edge weights (see Theorem 5.10) and so we propose another algorithm for finding a balanced weight assignment – one specifically designed for the setting when the input graph is polynomially sparse, but has exponential aspect ratio.

Suppose our input hypergraph, $H = (V, E, w)$ has edge weights in the range $[1, \exp(n)]$. We then divide hyperedges into weight categories such as $E_i = \{e \in E | w(e) \in [n^{10(i-1)}, n^{10i})\}$. We then bisect H into two hypergraphs H_1 and H_2, where H_1 contains all hyperedges in odd numbered categories, and H_2 all those in even numbered categories. This results in hyergraphs (H_1 and H_2) where hyperedges fall into extremely well-separated categories; so extremely in fact, that the weight of a hyperedge in a higher category (for example $e \in E_i \subseteq H_1$) has higher weight than all hyperedges of all lower categories *combined*, that is

$$w(e) \gg \sum_{e' \in E_{<i} \cap H_1} w(e').$$

We use this property to independently find weight assignments on H_1 and H_2. Informally, we go through the categories of hyperedges, from heaviest to lightest, resolving all instances of imbalance. We never return to a category once we moved on, and we prove that no amount of changes to the weight assignment of lower categories can disrupt the balance of a higher category, due to the huge discrepency in weights. For a more detailed and formal argument see Section 7.2.

4 Warm-Up: Ordinary Graphs

We begin by reproving the famous theorem of Spielman and Srivastava [SS11], which states that sampling edges of a graph with probability proportional to their effective resistance (and then reweighting appropriately) results in a spectral sparsifier with high probability. We prove a somewhat weaker version of the theorem, where we oversample by an $O(\epsilon^{-4} \log^3 n)$ factor, as opposed to the $\epsilon^{-2} \log n$ factor in the original. Another slight difference is that our version samples every edge independently, instead of sampling a predetermined number of edges with replacement in [SS11]. More recent proofs of the theorem of [SS11] that use the matrix Bernstein inequality as opposed to [RV07] also use the same distribution as ours.

Theorem 4.1 (A slightly weaker version of [SS11])**.** *Let $G = (V, F, z)$ be a weighted ordinary graph with n vertices and let $1/n \le \epsilon \le 1/2$. For every edge $f \in F$, let $p(f) = \min(1, \lambda \cdot z(f) \cdot R_G(f))$ for a sufficiently large factor $\lambda = \Theta(\epsilon^{-4} \log^3 n)$. Sample each edge $f \in F$ independently with probability $p(f)$, and give it weight $\widetilde{z}(f) = z(f)/p(f)$ if sampled. The resulting graph, $\widetilde{G} = (V, \widetilde{F}, \widetilde{z})$ is an ϵ-spectral sparsifier of G with probability at least $1 - O(\log n / n)$.*

The original proof of this theorem used a concentration bound for matrices [RV07] (later simplified to use the matrix Bernstein inequality) to prove that $x^\top L_{\widetilde{G}} x$ is close to its expectation simultaneously for all $x \in \mathbb{R}^n$, as required by Definition 2.2. This type of argument is difficult to adapt to hypergraph sparsification, because the extension of quadratic forms to hypergraphs

is highly non-linear. We thus present an alternative proof that uses more primitive techniques to bypass the reliance on linear algebra.

Proof of Theorem 4.1: By Definition 2.2, we must prove that for every $x \in \mathbb{R}^V$,

$$x^\top L_{\widetilde{G}} x \in (1 \pm \epsilon) \cdot x^\top L_G x. \tag{3}$$

We may assume without loss of generality that $x^\top L_G x = 1$. We denote the set of vectors x where this is satisfied as $S^G \subseteq \mathbb{R}^V$. Furthermore, we simplify notation by denoting L_G as L, and $L_{\widetilde{G}}$ as $\widetilde{L}$. Moreover, for any subset of edges $F' \subseteq F$, we denote the Laplacian of the subgraph of G corresponding to F' by $L_{F'}$, and similarly for the subgraph of $\widetilde{G}$ by $\widetilde{L}_{F'}$.

It is clear from the construction of $\widetilde{G}$ that

$$\mathbb{E}\left(x^\top \widetilde{L} x\right) = x^\top L x.$$

Therefore, we are in effect trying to prove the concentration of a random variable around its expectation in Equation (3). Indeed, for any specific x, Equation (3) holds with high probability by Chernoff bound (Theorem 2.10). (One can consider $x^\top L x$ as the sum of independent random variables of the form $\widetilde{z}(u, v) \cdot (x_u - x_v)^2$.)

In order to prove the concentration for all $x \in S^G$ simultaneously, we employ a net argument, where we "round" x to some vector from a finite set and apply a union bound on the rounded vectors. However, our rounding scheme is progressive and has $O(\log n)$ "levels" with increasingly finer resolution. Each x will then determine a partition of the edges into levels, and we will prove concentration for each rounded vector and each level (subset of edges), and then apply a union bound over all these choices.

The existence of these rounding functions is guaranteed by the following lemma, which we will prove in Section 4.1.

Lemma 4.2. *Let $G = (V, F, z)$ be a connected weighted graph. Then for every $i \in \mathbb{N}$ there exists a rounding function*

$$\varphi_i : S^G \to \mathbb{R}^V$$

such that for all $x \in S^G$, denoting $x^{(i)} := \varphi_i(x)$, we have:

1. *The image of φ_i is a finite set of cardinality $|\varphi_i(S^G)| \leq \exp\left(800 C \log n \cdot 2^i / \epsilon^2\right)$, where $C > 0$ is the absolute constant from Theorem 4.5.*

2. *For every edge $f = (u, v) \in F$ such that $\max\left((x_u - x_v)^2, (x_u^{(i)} - x_v^{(i)})^2\right) \geq 2^{-i} \cdot R_G(f)$,*

$$(x_u - x_v)^2 \in \left(1 \pm \frac{\epsilon}{7}\right) \cdot (x_u^{(i)} - x_v^{(i)})^2.$$

The second guarantee of Lemma 4.2 can be expressed in terms of the Laplacian of a single edge, resulting in the following corollary.

Corollary 4.3. *For a rounding function φ satisfying the guarantees of Lemma 4.2, and an edge $f = (u, v) \in F$ such that $\max\left((x_u - x_v)^2, (x_u^{(i)} - x_v^{(i)})^2\right) \geq 2^{-i} \cdot R_G(f)$,*

$$x^\top L_{\{f\}} x \in \left(1 \pm \frac{\epsilon}{7}\right) \cdot x^{(i)\top} L_{\{f\}} x^{(i)}.$$

Let us take a sequence of the rounding functions φ_i guaranteed by Lemma 4.2 for $i = 1, \ldots, I :=$ $\log_2(7n/\epsilon) \leq 3 \log n$. For each $x \in S^G$, it yields a sequence of rounded vectors $x^{(i)} = \varphi_i(x)$ for $i = 1, \ldots, I$. Furthermore, we use $x^{(i)}$ to define the subset of edges $F_i' \subseteq F$ by

$$F_i' := \left\{ f = (u, v) \in F \ \middle| \ \left(x_u^{(i)} - x_v^{(i)}\right)^2 \geq 2^{-i} \cdot R_G(f) \right\}.$$

That is, the second guarantee of Lemma 4.2 holds for φ_i on edges in F_i'. Finally, we use $\{F_i'\}_i$ to partition F as follows. Let the base case be $F_0 = F_0' := \{f \in F \mid p(f) = 1\}$, where we recall that $p(f) = \min(1, \lambda \cdot z(f) \cdot R_G(f))$. For each $i \in [I]$, let $F_i := F_i' \setminus \bigcup_{j=0}^{i-1} F_j'$, and finally let $F_{I+1} = F \setminus \bigcup_{i=0}^{I} F_i'$.

Thus we have partitioned F in such a way that the second guarantee of Lemma 4.2 applies to edges in F_i, with respect to φ_i. Furthermore, F_i are defined in terms of $x^{(i)}$ (and $x^{(j)}$ for $j < i$) instead of x, so that the number of possible sets F_i is finite, and bounded thanks to the first guarantee of Lemma 4.2.

We establish the following claim for later use.

Claim 4.4. *For all $i \in [I]$ and $f = (u, v) \in F_i$, we have*

$$(x_u^{(i)} - x_v^{(i)})^2 \leq 3 \cdot 2^{-i} \cdot R_G(f).$$

Proof. The second guarantee of Lemma 4.2 for φ_i applies to f, and thus $(x_u^{(i)} - x_v^{(i)})^2 \leq (x_u - x_v)^2 \cdot (1 - \epsilon/7)^{-1}$.

Consider first the case $i = 1$. By Fact 2.6 and since $x \in S^G$, we have $(x_u - x_v)^2 \leq R_G(f) \cdot x^\top L x = R_G(f)$, and we indeed get $(x_u^{(i)} - x_v^{(i)})^2 \leq R_G(f) \cdot (1 - \epsilon/7)^{-1} \leq 3 \cdot 2^{-1} \cdot R_G(f)$.

Now consider $i > 1$, and suppose towards contradiction that $(x_u^{(i)} - x_v^{(i)})^2 > 3 \cdot 2^{-i} \cdot R_G(f)$. Notice that the second guarantee of Lemma 4.2 also applies to f for φ_{i-1}, and thus $(x_u^{(i-1)} - x_v^{(i-1)})^2 \geq (x_u^{(i)} - x_v^{(i)})^2 \cdot (1 + \epsilon/7)^{-1} \cdot (1 - \epsilon/7) \geq 2^{-i+1} \cdot R_G(f)$. This implies that $f \in F_{i-1}'$, which contradicts the assumption $f \in F_i = F_i' \setminus \bigcup_{j=0}^{i-1} F_j'$. $\square$

We will consider each group of edges F_i separately, and prove that $x^\top \widetilde{L}_{F_i} x$ concentrates around its expectation, $x^\top L_{F_i} x$. More precisely, we will first prove concentration for every specific $(x^{(i)}, F_i)$, and then extend the concentration to all possibilities simultaneously via union bound. This is well-defined because each F_i depends on $x^{(1)}, \ldots, x^{(i)}$ but not directly on x.

Edges in F_0. By definition, every edge $f \in F_0$ has $p(f) = 1$, and thus $x^\top \widetilde{L}_{F_0} x$ is completely deterministic and equal to $x^\top L_{F_0} x$.

Edges in F_i for $i \in [I]$. Note that F_i is designed so that, by Corollary 4.3, for every edge $f \in F_i$ we have $\left| x^\top L_{\{f\}} x - x^{(i)\top} L_{\{f\}} x^{(i)} \right| \leq \epsilon/7 \cdot x^{(i)\top} L_{\{f\}} x^{(i)}$, and since $\widetilde{L}_{\{f\}}$ is a multiple of $L_{\{f\}}$ similarly have $\left| x^\top \widetilde{L}_{\{f\}} x - x^{(i)\top} \widetilde{L}_{\{f\}} x^{(i)} \right| \leq \epsilon/7 \cdot x^{(i)\top} \widetilde{L}_{\{f\}} x^{(i)}$. Informally, this allows us to prove concentration only for vectors $x^{(i)}$ instead of all x, and thus we next aim to bound the error

$$\left| x^{(i)\top} L_{F_i} x^{(i)} - x^{(i)\top} \widetilde{L}_{F_i} x^{(i)} \right|$$

for each $i \in [I]$ with high probability. It will then remain to bound the error introduced on the remaining edges (the ones in F_{I+1}).

Fix $i \in [I]$ and notice that over all possible vectors $x \in S^G$, there are only finitely many possible values for $(x^{(i)}, F_i)$. Therefore, we can focus on a single value of $x^{(i)}$ and F_i, and then use a union bound over all settings.

Let us therefore fix also $x^{(i)}$ and F_i. We will use Chernoff bounds to prove that with high probability, over the randomness of sampling edges to $\widetilde{G}$,

$$\left| x^{(i)\top} L_{F_i} x^{(i)} - x^{(i)\top} \widetilde{L}_{F_i} x^{(i)} \right| \leq \frac{\epsilon}{7I}. \tag{4}$$

Indeed, note that

$$x^{(i)\top} \widetilde{L}_{F_i} x^{(i)} = \sum_{f=(u,v)\in F_i} \widetilde{z}(f) \cdot (x_u^{(i)} - x_v^{(i)})^2,$$

where $\widetilde{z}(f)$ are independent random variables with expectation $\mathbb{E}(\widetilde{z}(f)) = z(f)$. Therefore, we can apply the Chernoff bound from Theorem 2.10 with $\{Z_i\}_i$ being $\widetilde{z}(f) \cdot (x_u - x_v)^2$ for each $f = (u,v) \in F_i$, and their sum being $Z = x^{(i)\top} \widetilde{L}_{F_i} x^{(i)}$ with $\mathbb{E}(Z) = x^{(i)\top} L_{F_i} x^{(i)}$. We need to set a as an upper bound on $\widetilde{z}(f) \cdot (x_u^{(i)} - x_v^{(i)})^2$. Observe that $\widetilde{z}(f)$ is maximal when f is sampled, in which case it equals $z(f)/p(f)$ where $p(f) = \lambda \cdot z(f) \cdot R_G(f)$, since $f \notin F_0$. We thus get, using Claim 4.4,

$$\forall f = (u,v) \in F_i, \qquad \frac{z(f) \cdot (x_u^{(i)} - x_v^{(i)})^2}{\lambda \cdot z(f) \cdot R_G(f)} = \frac{1}{\lambda} \cdot \frac{(x_u^{(i)} - x_v^{(i)})^2}{R_G(f)} \leq \frac{3 \cdot 2^{-i}}{\lambda} =: a.$$

We let $\delta := \epsilon/(14I)$, we can bound

$$x^{(i)\top} L_{F_i} x^{(i)} \leq \left(1 + \frac{\epsilon}{7}\right) \cdot x^\top L_{F_i} x \leq \left(1 + \frac{\epsilon}{7}\right) \cdot x^\top L x = 1 + \frac{\epsilon}{7} \leq 2 =: \mu.$$

(This is true for an arbitrary preimage $x \in \varphi_i^{-1}(x^{(i)})$.)

Finally, Theorem 2.10 implies

$$
\begin{aligned}
\mathbb{P}\left(\left| x^{(i)\top} L_{F_i} x^{(i)} - x^{(i)\top} \widetilde{L}_{F_i} x^{(i)} \right| \geq \frac{\epsilon}{7I} \right) &\leq 2\exp\left(-\frac{\delta^2 \mu}{3 \cdot a} \right) \\
&= 2\exp\left(-\frac{\frac{\epsilon^2}{196I^2} \cdot 2}{9 \cdot 2^{-i}/\lambda} \right) \\
&\leq 2\exp\left(-\frac{\epsilon^2 \cdot 2^i \cdot \lambda}{10000 \log^2(n)} \right) \\
&= 2\exp\left(-\frac{2000C \log n \cdot 2^i}{\epsilon^2} \right),
\end{aligned}
$$

where the last step by setting $\lambda = 2 \cdot 10^7 \cdot C \log^3 n / \epsilon^4$, where $C > 0$ is the absolute constant from Theorem 4.5.

We can now use a union bound to bound the probability that Equation (4) holds simultaneously for all values of $(x^{(i)}, F_i)$. F_i depends only on F_j' for $j \in [i]$, which in turn depend on $x^{(j)}$ for the same values of j. By the first guarantee of Lemma 4.2, the number of possible vectors $x^{(j)}$ is at most $\exp\left(800C \log n \cdot 2^j / \epsilon^2\right)$, where $C > 0$ is the absolute constant from Theorem 4.5. Therefore, the number of possible pairs $(x^{(i)}, F_i)$ is at most

$$\prod_{j=1}^{i} \exp\left(\frac{800C \log n \cdot 2^j}{\epsilon^2} \right) = \exp\left(\sum_{j=1}^{i} \frac{800C \log n \cdot 2^j}{\epsilon^2} \right) \leq \exp\left(\frac{1600C \log n \cdot 2^i}{\epsilon^2} \right).$$

11

Finally, the probability that Equation (4) *does not* hold simultaneously for all pairs $(x^{(i)}, F_i)$ is at most

$$\exp\left(\frac{1600C\log n \cdot 2^i}{\epsilon^2}\right) \cdot 2\exp\left(-\frac{2000C\log n \cdot 2^i}{\epsilon^2}\right) = 2\exp\left(-\frac{400C\log n \cdot 2^i}{\epsilon^2}\right) \leq \frac{1}{n}.$$

Edges in F_{I+1}. First we show that for any $x \in S^G$ and any edge $f = (u,v) \in F_{I+1}$ we have that $(x_u - x_v)^2 \leq \epsilon \cdot R_G(f)/(6n)$. Suppose for contradiction that this is not the case. Then the second guarantee of Lemma 4.2 applies and $(x_u^{(I)} - x_v^{(I)})^2 \geq (x_u - x_v)^2 \cdot (1-\epsilon/7) \geq \epsilon \cdot R_G(e)/(6n) \cdot (1-\epsilon/7) \geq \epsilon \cdot R_G(e)/(7n)$. Therefore $f \in F_I'$, which contradicts the assumption $f \in F_{I+1}$. (Here we used that I was defined to be $\log_2(7n/\epsilon)$.)

Next, we would like to bound $\left|x^\top \widetilde{L}_{F_{I+1}} x - x^\top L_{F_{I+1}} x\right|$ by showing that both terms are small. First,

$$x^\top L_{F_{I+1}} x = \sum_{f=(u,v)\in F_{I+1}} z(f) \cdot (x_u - x_v)^2 \; \leq \; \sum_{f\in F_{I+1}} z(f) \cdot \epsilon \cdot \frac{R_G(f)}{6n} \leq \frac{\epsilon}{6n} \cdot \sum_{f\in F} z(f) \cdot R_G(f) \; \leq \frac{\epsilon}{6},$$

where the last inequality uses Fact 2.9. Second, we start similarly,

$$x^\top \widetilde{L}_{F_{I+1}} x = \sum_{f=(u,v)\in F_{I+1}} \widetilde{z}(f) \cdot (x_u - x_v)^2 \; \leq \; \sum_{f\in F_{I+1}} \widetilde{z}(f) \cdot \epsilon \cdot \frac{R_G(f)}{6n} \leq \frac{\epsilon}{6n} \cdot \sum_{f\in F} \widetilde{z}(f) \cdot R_G(f),$$

and ideally we would like to show that $\sum \widetilde{z}(f) \cdot R_G(f) \leq 2n$. This is not always true, but it is a random event, independent of the choice of x, and can be shown to hold with high probability using our Chernoff bound from Theorem 2.10. Indeed, $\widetilde{z}(f) \cdot R_G(f)$ are independent random variables with maximum value when f is sampled, in which case $\widetilde{z}(f) = z(f)/p(f)$, and thus

$$\widetilde{z}(f) \cdot R_G(f) \leq \frac{z(f) \cdot R_G(f)}{p(f)} = \frac{z(f) \cdot R_G(f)}{\min(1, \lambda \cdot z(f) \cdot R_G(f))} = \max\left(z(f) \cdot R_G(f), 1/\lambda\right) \leq 1 =: a,$$

where the last inequality uses Fact 2.8. We apply Theorem 2.10 by setting $\delta := 1$ and $\mu := n$ (which we may do by Fact 2.9), and obtain

$$\mathbb{P}\left(\sum_{f\in F} \widetilde{z}(f) \cdot R_G(f) \geq 2n\right) \leq 2\exp\left(-\frac{n}{3}\right).$$

Therefore, with probability at least $1 - O(1/n)$,

$$\left|x^\top \widetilde{L}_{F_{I+1}} x - x^\top L_{F_{I+1}} x\right| \leq \frac{\epsilon}{2}. \tag{5}$$

Putting everything together. By the above derivations, Equation (4) holds for all i and all $(x^{(i)}, F_i)$ simultaneously, as well as Equation (5) holds with probability at least $1 - O(\log n/n)$. Assuming henceforth that this high probability event occurs, we shall deduce that Equation (3) holds for all $x \in S^G$. Indeed, by the triangle inequality and Equation (5),

$$\left|x^\top \widetilde{L} x - x^\top L x\right| \leq \sum_{i=0}^{I+1} \left|x^\top \widetilde{L}_{F_i} x - x^\top L_{F_i} x\right| \leq 0 + \sum_{i=1}^{I} \left|x^\top \widetilde{L}_{F_i} x - x^\top L_{F_i} x\right| + \frac{\epsilon}{2}.$$

12

Now for each $i \in [I]$, we can approximate terms involving x by $x^{(i)}$ and vice versa, formalized by the aforementioned fact that for every $(u, v) \in F_i$ we have $|(x_u - x_v)^2 - (x_u^{(i)} - x_v^{(i)})^2| \leq \epsilon/7 \cdot (x_u^{(i)} - x_v^{(i)})^2$ (see the second condition of Lemma 4.2 and the definition of $F_i' \subseteq F_i$), and get

$$\left| x^\top \widetilde{L}_{F_i} x - x^\top L_{F_i} x \right| \leq \left| x^\top \widetilde{L}_{F_i} x - x^{(i)\top} \widetilde{L}_{F_i} x^{(i)} \right| + \left| x^{(i)\top} \widetilde{L}_{F_i} x^{(i)} - x^{(i)\top} L_{F_i} x^{(i)} \right| + \left| x^{(i)\top} L_{F_i} x^{(i)} - x^\top L_{F_i} x \right|$$

$$\leq \frac{\epsilon}{7} \cdot x^{(i)\top} \widetilde{L}_{F_i} x^{(i)} + \left| x^{(i)\top} \widetilde{L}_{F_i} x^{(i)} - x^{(i)\top} L_{F_i} x^{(i)} \right| + \frac{\epsilon}{7} \cdot x^{(i)\top} L_{F_i} x^{(i)}$$

now we use the triangle inequality,

$$\leq \frac{\epsilon}{7} \cdot x^{(i)\top} L_{F_i} x^{(i)} + \left(1 + \frac{\epsilon}{7} \right) \cdot \left| x^{(i)\top} \widetilde{L}_{F_i} x^{(i)} - x^{(i)\top} L_{F_i} x^{(i)} \right| + \frac{\epsilon}{7} \cdot x^{(i)\top} L_{F_i} x^{(i)}$$

and now we crucially use Equation (4),

$$\leq \left(1 + \frac{\epsilon}{7} \right) \cdot \frac{\epsilon}{7I} + \frac{2\epsilon}{7} \cdot \left(1 - \frac{\epsilon}{7} \right)^{-1} \cdot x^\top L_{F_i} x$$

$$\leq \frac{\epsilon}{6I} + \frac{2\epsilon}{6} \cdot x^\top L_{F_i} x.$$

Substituting this into our previous bound, we obtain

$$\left| x^\top \widetilde{L} x - x^\top \widetilde{L} x \right| \leq \sum_{i=1}^{I} \left(\frac{\epsilon}{6I} + \frac{\epsilon}{3} \cdot x^\top L_{F_i} x \right) + \frac{\epsilon}{2} \leq \frac{\epsilon}{6} + \frac{\epsilon}{3} \cdot x^\top L x + \frac{\epsilon}{2} = \epsilon \cdot x^\top L x,$$

where the last equality uses $x^\top L x = 1$. This completes the proof of Theorem 4.1. $\qquad \square$

4.1 Proof of Lemma 4.2

To prove Lemma 4.2, we use the following Theorem:

Theorem 4.5 (Theorem VI.1 of [AK17]). *Let $a_1, \ldots, a_m \in \mathbb{R}^n$ be vectors of norm at most 1 and let $\eta \in (0, 1)$. Then, over all vectors $y \in \mathbb{R}^n$ of norm at most 1, the number of possible values of the "rounded vector"*

$$\left(\left\lfloor \frac{\langle a_1, y \rangle}{\eta} \right\rfloor, \left\lfloor \frac{\langle a_2, y \rangle}{\eta} \right\rfloor, \ldots, \left\lfloor \frac{\langle a_k, y \rangle}{\eta} \right\rfloor \right)$$

is at most $\exp\left(\frac{C \log m}{\eta^2} \right)$ for some absolute constant $C > 0$.

Remark 4.6. *In fact, the original theorem (Theorem 6.1 in [AK17]) is stated with stronger requirements on m and η, and a stronger consequence. However, we can easily get the weaker upper bound of $\exp(O(\log m/\eta^2))$ stated in Theorem 4.5 of this paper, by setting the variables appropriately: $\varepsilon := \eta$, $n := \max(m, 1/\eta^2)$, and $k := n$, where the left hand side always represents their variable names and the right hand side ours.*

Proof of Lemma 4.2: We use the idea of resistive embedding introduced in []. Note that $L = L_G$ is a positive semidefinite matrix, and we denote by $L^{+/2}$ the square root of its Moore-Penrose pseudo-inverse. For each (unordered) vertex pair (u, v), let $b_{u,v} \in \mathbb{R}^V$ be the vector with all zero coordinates, except for the coordinates associated with u and v, which are 1 and -1 (ordered arbitrarily). With each vertex pair (u, v), we associate the vector

$$a_{u,v} = \frac{L^{+/2} b_{u,v}}{\|L^{+/2} b_{u,v}\|_2}.$$

13

Furthermore, we associate with each $x \in S^G$ the vector $y_x = L^{1/2}x$.

We can then apply Theorem 4.5 to $\{a_{u,v} \mid (u,v) \in \binom{V}{2}\}$ and all possible y_x, setting $\eta = \epsilon \cdot 2^{-i/2}/20$. Indeed, $a_{u,v}$ is normalized by definition, and also y_x is normalized because $x \in S^G$ and thus

$$\|y_x\|_2^2 = x^\top L^{1/2} L^{1/2} x = x^\top L x = 1.$$

For each possible value of the rounded vector

$$\left(\left\lfloor \left|\frac{\langle a_{u,v}, y_x \rangle}{\eta}\right| \right\rfloor\right)_{(u,v) \in \binom{V}{2}}$$

choose a representative $x \in S^G$, and let φ_i map each $x \in S^G$ to its representative (i.e., with the same rounded vector). Then by Theorem 4.5, the image of φ_i is of size $|\varphi_i(S^G)| \leq \exp\left(800C \log n \cdot 2^i/\epsilon^2\right)$, as claimed. Recall that we denote $\varphi_i(x)$ by $x^{(i)}$; then

$$\left(\left\lfloor \left|\frac{\langle a_{u,v}, y_x \rangle}{\eta}\right| \right\rfloor\right)_{(u,v) \in \binom{V}{2}} = \left(\left\lfloor \left|\frac{\langle a_{u,v}, y_{x^{(i)}} \rangle}{\eta}\right| \right\rfloor\right)_{(u,v) \in \binom{V}{2}}.$$

It follows that for all $f = (u,v) \in F$ and all $x \in S^G$,

$$|\langle a_{u,v}, y_x \rangle - \langle a_{u,v}, y_{x^{(i)}} \rangle| \leq \eta. \tag{6}$$

Furthermore, $b_{u,v}$ is perpendicular to the null-space of L (which is spanned by the all-ones vector because G is connected), thus $L^{1/2} L^{+/2} b_{u,v} = b_{u,v}$ and

$$\langle a_{u,v}, y_x \rangle^2 = \frac{\left(x^\top L^{1/2} L^{+/2} b_{u,v}\right)^2}{b_{u,v}^\top L^{+/2} L^{+/2} b_{u,v}} = \frac{\left(x^\top b_{u,v}\right)^2}{b_{u,v}^\top L^+ b_{u,v}} = \frac{(x_u - x_v)^2}{R_G(u,v)}. \tag{7}$$

To prove the second guarantee of Lemma 4.2, let $f = (u,v) \in F$ and $x \in S^G$ and consider first the case $\left(x_u^{(i)} - x_v^{(i)}\right)^2 \geq 2^{-i} \cdot R_G(f)$, which by (7) is equivalent to $\langle a_{u,v}, y_{x^{(i)}} \rangle^2 \geq 2^{-i}$. This means that the absolute error bound η in Equation (6) implies a relative error bound, namely,

$$|\langle a_{u,v}, y_x \rangle - \langle a_{u,v}, y_{x^{(i)}} \rangle| \leq \eta = \frac{\epsilon \cdot 2^{-i/2}}{20} \leq \frac{\epsilon}{20} \cdot |\langle a_{u,v}, y_{x^{(i)}} \rangle|.$$

The other case $(x_u - x_v)^2 \geq 2^{-i} \cdot R_G(f)$ is similar up to constants; by (7), this case is equivalent to $\langle a_{u,v}, y_x \rangle^2 \geq 2^{-i}$, and thus

$$|\langle a_{u,v}, y_x \rangle - \langle a_{u,v}, y_{x^{(i)}} \rangle| \leq \eta = \frac{\epsilon \cdot 2^{-i/2}}{20} \leq \frac{\epsilon}{20} \cdot |\langle a_{u,v}, y_x \rangle|,$$

which implies

$$|\langle a_{u,v}, y_x \rangle - \langle a_{u,v}, y_{x^{(i)}} \rangle| \leq \frac{\epsilon}{20} \cdot \left(1 - \frac{\epsilon}{20}\right)^{-1} |\langle a_{u,v}, y_x \rangle| \leq \frac{\epsilon}{16} \cdot |\langle a_{u,v}, y_{x^{(i)}} \rangle|.$$

Now in both cases,

$$\left|\langle a_{u,v}, y_x \rangle^2 - \langle a_{u,v}, y_{x^{(i)}} \rangle^2\right| = |\langle a_{u,v}, y_x \rangle - \langle a_{u,v}, y_{x^{(i)}} \rangle| \cdot |\langle a_{u,v}, y_x \rangle + \langle a_{u,v}, y_{x^{(i)}} \rangle|$$

$$\leq \frac{\epsilon}{16} \cdot |\langle a_{u,v}, y_{x^{(i)}} \rangle| \cdot \left(2 + \frac{\epsilon}{16}\right) \cdot |\langle a_{u,v}, y_{x^{(i)}} \rangle|$$

$$\leq \frac{\epsilon}{7} \cdot \langle a_{u,v}, y_{x^{(i)}} \rangle^2.$$

Using (7) and scaling by $R_G(u,v)$, we can write this as $(x_u - x_v)^2 \in (1 \pm \epsilon/7) \cdot (x_u^{(i)} - x_v^{(i)})^2$, which completes the proof of Lemma 4.2. $\square$

5 γ-Balanced Weight Assignments

Our strategy for generalizing the techniques of Section 4 to hypergraphs is similar to that of [CKN20]. Intuitively, we wish to replace each hyperedge of the input hypergraph with a weighted clique in such a way that the "importance" of each edge in the same clique is roughly the same. However, our measure of importance is effective resistance, whereas in [CKN20] it is the *strength* of the edge (see [BK15]), which is a measure particularly useful to cut sparsification. Specifically, we use the following definition.

Definition 5.1. *Given a hypergraph $H = (V, E, w)$, a weight assignment of H is a weighted (ordinary) graph $G = (V, F, z)$ such that*

- *F is the multiset $\bigcup_{e \in E} F_e$, where F_e is a set of edges forming a clique on the support of e*

- *$\sum_{f \in F_e} z(f) = w(e)$.*

Note that this definition allows for parallel edges in G.

Moreover, if G satisfies

$$\gamma \cdot \min_{g \in F_e:\ z(g)>0} R_G(g) \geq \max_{f \in F_e} R_G(f)$$

for $\gamma > 1$, then we call it γ-balanced.

The goal of this section is to show the existence of constant-balanced weight assignments for all weighted hypergraphs, and to define an efficient algorithm that outputs such a weight assignment.

5.1 Spanning Tree Potential

In order to show the existence (and give an efficient construction) of γ-balanced weight assignments we introduce the concept of spanning tree potentials for weighted ordinary graphs.

Definition 5.2 (Spanning tree potential, ST-potential for short)**.** *For a connected weighted graph $G = (V, F, z)$ let $\mathbb{T}(G)$ be the set of all spanning trees of G. Then we define the spanning tree potential of G as*

$$\Psi(G) = \log \left[\sum_{T \in \mathbb{T}(G)} \prod_{f \in T} z(f) \right]. \tag{8}$$

Remark 5.3. *Note that the value of ST-potential stays the same after replacing parallel edges f_1 and f_2 with a single edge f of weight $z(f) := z(f_1) + z(f_2)$.*

We formalize the concept of edge updates to graphs and see how those updates affect ST-potential.

Definition 5.4. *If $G = (V, F, z)$ is a weighted graph, $\lambda \in \mathbb{R}$, and $f \in \binom{V}{2}$, then $G + \lambda \cdot f$ is the weighted graph (V, F', z'), where the weight of f is increased by λ. Formally,*

- *$F' = F$, $z'(f) = z(f) + \lambda$, and $z'(g) = z(g)$ for all $g \neq f$, if $f \in F$,*

- *or $F' = F + \{f\}$, $z'(f) = \lambda$, and $z'(g) = z(g)$ for all $g \neq f$, if $f \notin F$.*

Note that the definition applies to $\lambda < 0$ as well, but in this case f must be present in the graph with weight at least $|\lambda|$ in order for $G + \lambda \cdot f$ to be valid.

Lemma 5.5. *For any weighted graph G, $\lambda \in \mathbb{R}$ and $f \in \binom{V}{2}$ such that $G + \lambda \cdot f$ is well defined, we have*

$$\Psi(G + \lambda \cdot f) = \Psi(G) + \log\left(\lambda R_G(f) + 1\right), \tag{9}$$

Proof. We use the famous result that if we sample a spanning tree $\mathcal{T}$ randomly from $\mathbb{T}(G)$ such that

$$\mathbb{P}(\mathcal{T} = T) \propto \prod_{g \in T} z(g),$$

then the marginal probability $\mathbb{P}(f \in T)$ is $z(f) \cdot R(f)$ for all $f \in F$ (see, e.g., [Lov93]). Let $\mathcal{T}$ be a random variable drawn from such a distribution. By the definition of the distribution of $\mathcal{T}$ we have that

$$\mathbb{P}(f \in \mathcal{T}) = \sum_{T \in \mathbb{T}(G):T \ni f} \mathbb{P}(\mathcal{T} = T) = \frac{\sum_{T \in \mathbb{T}(G):T \ni f} \prod_{g \in T} z(g)}{\sum_{T \in \mathbb{T}(G)} \prod_{g \in T} z(g)} \tag{10}$$

Therefore, we have for the weight function z' of $G + \lambda \cdot f$,

$$\Psi(G + \lambda \cdot f) = \log\left[\sum_{T \in \mathbb{T}(G')} \prod_{g \in T} z'(g)\right]$$

$$= \log\left[\sum_{T \in \mathbb{T}(G)} \prod_{g \in T} z(g) \cdot \left(1 + \frac{\lambda}{z(f)}\right)^{\mathbb{1}(g=f)}\right]$$

$$= \log\left[\sum_{T \in \mathbb{T}} \left(1 + \frac{\lambda}{z(f)}\right)^{\mathbb{1}(f \in T)} \prod_{g \in T} z(g)\right]$$

$$= \log\left[\sum_{T \in \mathbb{T}(G)} \prod_{g \in T} z(g) + \frac{\lambda}{z(f)} \cdot \sum_{T \in \mathbb{T}(G):T \ni f} \prod_{g \in T} z(g)\right].$$

We can transform the second term by Equation (10) to continue:

$$= \log\left[\sum_{T \in \mathbb{T}(G)} \prod_{g \in T} z(g) + \frac{\lambda}{z(f)} \cdot \mathbb{P}(f \in \mathcal{T}) \cdot \sum_{T \in \mathbb{T}(G)} \prod_{g \in T} z(g)\right]$$

$$= \log\left[\sum_{T \in \mathbb{T}(G)} \prod_{g \in T} z(g)\right] + \log\left(\frac{\lambda}{z(f)} \cdot \mathbb{P}(f \in \mathcal{T}) + 1\right)$$

$$= \Psi(G) + \log(\lambda R_G(f) + 1)$$

as claimed. Note that the above calculation is legitimate for positive λ as well as negative. $\square$

5.2 Existence of γ-Balanced Weight Assignments

We will now use ST-potential to construct a γ-balanced weight assignment of an arbitrary hypergraph. We can analyze a simple greedy algorithm, which identifies edge pairs contradicting the γ-balancedness condition in Definition 5.1, and simply shifts weight from the one with smaller effective resistance to the one with the larger effective resistance (see Algorithm 1). One can show that such an step can be designed to only increase the ST-potential of a weight assignment, and thus the algorithm eventually terminates, returning a γ-balanced weight assignment.

Algorithm 1 Algorithm for constructing a γ-balanced weight assignment.

1: **procedure** GREEDYBALANCING($H = (V, E, w), \gamma$)
2:     For all $e \in E$ and for all $f \in F_e$, initialize $z(f)$ to $w(e)/\binom{|e|}{2}$
3:     $G \leftarrow (V, \bigcup_{e \in E} F_e, z)$
4:     **while** G is not a γ-balanced weight assignment of H **do**
5:         Select $e \in E$, and $f, g \in F_e$, such that $R_G(f) > \gamma \cdot R_G(g)$ and $z(g) > 0$
6:         $\lambda \leftarrow \min\left(z(g), (\gamma - 1)/(2\gamma \cdot R_G(g))\right)$
7:         $z(f) \leftarrow z(f) + \lambda$
8:         $z(g) \leftarrow z(g) - \lambda$
9:     **return** G

The following tells how much the effective resistance of an edge changes by updating the weight of another edge. Although the proof is simple and this result is already known, we include the proof for completeness.

Lemma 5.6. *If $G = (V, F, z)$ is a weighted graph, let $\lambda \in \mathbb{R}$, and $f \in \binom{V}{2}$, then for any $g \in \binom{V}{2}$*

$$R_{G+\lambda \cdot f}(g) = R_G(g) - \frac{\lambda \cdot \left(b_g^\top L_G^+ b_f\right)^2}{1 + \lambda \cdot R_G(f)}.$$

Proof. Note that

$$L_{G+\lambda \cdot f} = L_G + \lambda b_f b_f^\top.$$

Therefore, by the Sherman-Morrison formula for Moore-Penrose pseudoinverse (see for example [Mey73]), we can expand the formula for the effective resistance of g:

$$R_{G+\lambda \cdot f}(g) = b_g^\top L_{G+\lambda \cdot f}^+ b_g = b_g^\top \left(L_G + \lambda b_f b_f^\top\right)^+ b_g = b_g^\top \left(L_G^+ - \frac{L_G^+ b_f \lambda b_f^\top L_G^+}{1 + \lambda b_f^\top L_G^+ b_f}\right) b_g$$

$$= b_g^\top L_G^+ b_g - \frac{\lambda b_g^\top L_G^+ b_f b_f^\top L_G^+ b_g}{1 + \lambda b_f^\top L_G^+ b_f} = R_G(g) - \frac{\lambda \cdot \left(b_g^\top L_G^+ b_f\right)^2}{1 + \lambda \cdot R_G(f)}. \qquad \square$$

Lemma 5.7. *Let $G = (V, F, z)$ be a weighted graph and let $\gamma > 1$. Let f, g be two edges in F such that $R_G(f) > \gamma \cdot R_G(g)$. Then for any $\lambda \leq z(g)$, shifting λ weight from g to f results in an increase of at least*

$$\log\left(1 + \lambda\gamma \cdot R_G(g) - \lambda \cdot R_G(g) - \lambda^2\gamma \cdot R_G(g)^2\right)$$

in the ST-potential of G.

Proof. We simply apply the update formula for ST-potential (Lemma 5.5) twice, along with Lemma 5.6. For simplicity, we use t_{fg} to denote $b_f^\top L_G^+ b_g$. Then the increase in ST-potential is

$$\log\left(\lambda \cdot R_G(f) + 1\right) + \log\left(-\lambda \cdot R_{G+\lambda \cdot f}(g) + 1\right)$$

$$= \log\left(\lambda \cdot R_G(f) + 1\right) + \log\left(-\lambda \cdot \left(R_G(g) - \frac{\lambda t_{fg}^2}{1 + \lambda R_G(f)}\right) + 1\right)$$

$$= \log\left[\left(\lambda \cdot R_G(f) + 1\right) \cdot \left(-\lambda \cdot \left(R_G(g) - \frac{\lambda t_{fg}^2}{1 + \lambda R_G(f)}\right) + 1\right)\right]$$

17

$$= \log \left(1 + \lambda \cdot R_G(f) - \lambda \cdot R_G(g) - \lambda^2 \cdot R_G(f) R_G(g) + \lambda^2 t_{fg}^2 \right)$$
$$\geq \log \left(1 + \lambda \gamma \cdot R_G(g) - \lambda \cdot R_G(g) - \lambda^2 \gamma \cdot R_G(g)^2 \right),$$

as claimed. $\square$

Theorem 5.8. *For $\gamma > 1$, Algorithm 1 terminates and returns a γ-balanced weight assignment of H.*

Proof. It is clear by the condition of the while-loop that if Algorithm 1 terminates, it returns a γ-balanced weight assignment. Also, the condition $\sum_{f \in F_e} z(f) = w(e)$ is never violated. Therefore, in Line 1, there must indeed always be some $e \in E$ and some $f, g \in F_e$ such that $R_G(f) > \gamma \cdot R_G(g)$ and $z(g) > 0$, otherwise G would already be γ-balanced. It remains to prove that Algorithm 1 always terminates.

Let us examine the evolution of $\Psi(G)$ throughout the algorithm. First, note that G never becomes disconnected, and hence $\Psi(G)$ always remains defined. Indeed, in order for G to become disconnected, we would have to set λ to $z(g)$ for a *bridge* g. However, if g is a bridge, $R_G(g) = 1/z(g)$ by Fact 2.8, and λ is set instead to $(\gamma - 1)z(g)/(2\gamma) < z(g)$.

In each iteration of the while-loop (which we call a step) we move λ weight from some edge g to some other edge f. By Lemma 5.7 this results in a change of

$$\log \left(1 + \lambda \gamma \cdot R_G(g) - \lambda \cdot R_G(g) - \lambda^2 \gamma \cdot R_G(g)^2 \right).$$

Here we distinguish between the following two cases: If $\lambda = (\gamma - 1)/(2\gamma \cdot R_G(g))$ exactly, then the increase in $\Psi(G)$ is at least $\log(1 + (\gamma-1)^2/(2\gamma) - (\gamma-1)^2/(4\gamma)) \geq \log(1 + (\gamma-1)^2/(4\gamma)) =: c_\gamma$. On the other hand, if $\lambda = z(g) \leq (\gamma - 1)/(2\gamma \cdot R_G(g))$ then the increase in $\Psi(G)$ is at least $\log(1 + \lambda \gamma \cdot R_G(g) - \lambda \cdot R_G(g) - \lambda(\gamma - 1) \cdot R_G(g)/2) > \log(1) = 0$.

Overall there are two possibilities each step:

1. $\Psi(G)$ increases by at least $c_\gamma > 0$,

2. or $z(g)$ becomes 0, and $\Psi(G)$ increases by a positive amount.

Let the initial setting of G (before the while-loop) be G_0. Let G_∞ be the complete graph on V with uniform edge weights of $\sum_{e \in E} w(e)$ on each edge. Since G always satisfies

$$\sum_{f \in F} z(f) = \sum_{e \in E} \sum_{f \in F_e} z(f) = \sum_{e \in E} w(e),$$

$\Psi(G)$ will always be less than $\Psi(G_\infty)$ by monotonicity of Ψ. Thus, there can be at most $(\Psi(G_\infty) - \Psi(G_0))/c_\gamma$ steps of type 1. Therefore, after a certain point, there can only be steps of type 2; we focus on this stage of the algorithm.

We further categorize steps of type 2 based on the initial weight of f:

2a. The initial weight of f is greater than 0,

2b. the initial weight of f is exactly 0.

Steps of type 2a increase the total number of edges of weight exactly 0, since by definition, neither f nor g starts out with weight 0, but $z(g)$ becomes 0. On the other hand, steps of type 2b do not decrease the total number of edges of weight 0. Therefore, after a certain point, there can only be steps of type 2b; we focus on this stage of the algorithm.

At this point, the set of all edge weight values, that is

$$\bigcup_{f \in F} \{z(f)\},$$

remains unchanged. Indeed at every step we simply switch the values of $z(f)$ and $z(g)$. Therefore, there are only a finite number of possible states for G to be in. None of these can be repeated, as $\Psi(G)$ increases by a positive amount after each step, and therefore, Algorithm 1 must terminate, returning a γ-balanced weight assignment of H. □

This proves the existence of γ-balanced weight assignment, which suffices to show the existence of nearly-linear-sized spectral sparsifiers (as we will see in Section 6). Unfortunately, the above proof shows no bound on the running time of Algorithm 1 beyond $2^{O(m)}$.

5.3 Polynomial-Time Construction

In this section, we introduce a relaxation of the concept of γ-balanced weight assignment. This will still be sufficient to get spectral sparsifiers of nearly linear size, while also allowing the greedy algorithm to terminate in a polynomial number of steps.

Definition 5.9. *For $0 < \eta \leq 1$ and $\gamma > 1$, an η-approximate γ-balanced weight assignment of $H = (V, E, w)$ is defined exactly as above in Definition 5.1, except with the final condition relaxed to*

$$\gamma \cdot \min_{g \in F_e:\ z(g) \geq \eta \cdot w(e)} R_G(g) \geq \max_{f \in F_e} R_G(f).$$

That is, we allow edges not only of weight 0, but also of weight less than $\eta \cdot w(e)$ to be small outliers in terms of effective resistance.

We modify Algorithm 1 to search for approximate γ-balanced weight assignments.

Algorithm 2 Algorithm for constructing an η-approximate γ-balanced weight assignment.

1: **procedure** GREEDYAPPROXBALANCING($H = (V, E, w), \gamma, \eta$)
2:     For all $e \in E$ and for all $f \in F_e$, initialize $z(f)$ to $w(e)/\binom{|e|}{2}$
3:     $G \leftarrow (V, \bigcup F_e, z)$
4:     **while** G is not an η-approximate γ-balanced weight assignment of H **do**
5:         Select $e \in E$, and $f, g \in F_e$, such that $R_G(f) > \gamma \cdot R_G(g)$ and $z(g) > \eta \cdot w(e)$
6:         $\lambda \leftarrow \min\left(z(g), 1/(2R_G(g))\right)$
7:         $z(f) \leftarrow z(f) + \lambda$
8:         $z(g) \leftarrow z(g) - \lambda$
9:     **return** G

Theorem 5.10. *Let $H = (V, E, w)$ be a weighted hypergraph Then for $\gamma \geq 4$ and $\eta > 0$, Algorithm 2 terminates* within $M/(\eta w_{\min}) \cdot \operatorname{poly}\left(n \log(M/w_{\min})\right)$ rounds *and returns an η-approximate γ-balanced weight assignment for H, where $w_{\min} := \min_{e \in E} w(e)$ and $M := \sum_{e \in E} w(e)$.*

Proof. It is clear by the condition of the while-loop that if Algorithm 2 terminates, then it returns an η-approximate γ-balanced weight assignment. Also, the condition $\sum_{f \in F_e} z(f) = w(e)$ is never violated. Therefore, in Line 2, there must indeed always be some $e \in E$ and some $f, g \in F_e$ such that $R_G(f) > \gamma$ and $z(g) > \eta \cdot w(e)$, otherwise G would already be η-approximately γ-balanced. It remains to prove that Algorithm 2 terminates within $M/(\eta w_{\min}) \cdot \operatorname{poly}\left(n \log(M/w_{\min})\right)$ rounds.

Let G_0 be the starting graph of Algorithm 2, and G_∞ be the complete graph on V with uniform edge weights of M. Since $\Psi(G_\infty)$ is always greater than $\Psi(G)$ at every moment in the algorithm, we can upper bound the total increase in ST-potential throughout the algorithm by $\Psi(G_\infty) - \Psi(G_0)$. Let us upper bound $\Psi(G_\infty) - \Psi(G_0)$. To this end we will produce a sequence of updates to G_0 resulting in G_∞, and we will upper bound the contribution of each update.

Since $\min_{e \in E} w(e) = w_{\min}$, $\min_{f \in F} z(f) \geq w_{\min}/n^2$ in G_0. Furthermore, since H is connected, G_0 is connected as well and must contain a spanning tree with edges of weight at least $w_{\min}/n^2$ and therefore $\min_{u,v \in V} R_{G_0}(u,v) \leq n/(w_{\min}/n^2) = n^3/w_{\min}$. To transform G_0 to G_∞, we simply add to each vertex pair (u,v) a sufficient amount of weight to make $w(u,v) = M$. This is an update of $+\lambda \cdot (u,v)$ with $\lambda \leq M$ and $R(u,v) \leq n^3/w_{\min}$, which contributes by at most $\log(Mn^3/w_{\min} + 1)$ to Ψ by Lemma 5.5. The total contribution of all such updates on the way from G_0 to G_∞ is at most

$$\sum_{u,v \in V} \log\left(\frac{Mn^3}{w_{\min}} + 1\right) = \mathrm{poly}\left(n \log\left(\frac{M}{w_{\min}}\right)\right).$$

We now lower bound the minimum increase of Ψ after each step of Algorithm 2. We are able to do this thanks to the modification in Definition 5.9 of *approximate γ-balanced weight assignment*.

Due to Lemma 5.7, the contribution of each update to the potential $\Psi(G)$ is at least

$$\log\left(1 + \lambda\gamma \cdot R_G(g) - \lambda \cdot R_G(g) - \lambda^2\gamma \cdot R_G(g)^2\right)$$

At each step, λ is set to either $z(g)$ or $1/(2R_G(g))$. If $\lambda = 1/(2R_G(g))$, the increase in $\Psi(G)$ is at least $\log(1 + (\gamma - 1)/2 - \gamma/4) \geq \log(5/4)$. On the other hand, if $\lambda = z(g) \leq 1/(2R_G(g))$, then the increase in $\Psi(G)$ is at least

$$\begin{aligned}
\log(1 + \lambda\gamma \cdot R_G(g) - \lambda \cdot R_G(g) - \lambda\gamma \cdot R_G(g)/2) &\geq \log(1 + \lambda \cdot R_G(g)) \\
&= \log(1 + z(g) \cdot R_G(g)) \\
&\geq \log(1 + \eta \cdot w_{\min} \cdot R_G(g)).
\end{aligned}$$

To lower bound this, note that for all g

$$R_G(g) \geq R_{G_\infty}(g) = \frac{2}{Mn}.$$

This gives us that after each round of the algorithm $\Psi(G)$ increases by at least

$$\log(1 + \eta \cdot w_{\min} \cdot R_G(g)) \geq \frac{2 \cdot 2\eta w_{\min}/(nM)}{2 + 2\eta w_{\min}/(nM)} \geq \frac{4\eta w_{\min}/(nM)}{4} = \frac{\eta w_{\min}}{nM},$$

where we used $\log(1 + x) \geq (2x)/(2 + x)$ for $x \geq 0$ in the first inequality and we assumed n is sufficiently large in the second inequality. Thus the algorithm takes at most $M/(\eta w_{\min}) \cdot \mathrm{poly}\left(n \log(M/w_{\min})\right)$ steps. $\qquad\square$

6 Hypergraph Sparsification

In this section, we prove the existence of a spectral sparsifier with a nearly linear number of hyperedges, that is, the first part of Theorem 1. We discuss efficient construction of spectral sparsifier in Section 7.

To construct a spectral sparsifier, we first produce a $1/n^2$-approximate γ-balanced weight assignment for the input hypergraph, where $\gamma \geq 4$. We can use Algorithm 2 for this. We then assign

20

to each hyperedge e importance equal to the maximum effective resistance in F_e (see Definition 5.1). We then perform typical importance sampling on the hyperedges, oversampling by a factor of λ (to be set later). A formal description is given in Algorithm 3.

Algorithm 3 ϵ-spectral sparsification for a hypergraph, using importance sampling.

1: **procedure** SPARSIFICATION($H = (V, E, w), G = (V, F, z), \epsilon, \lambda$)
2:    $\widetilde{H} = (V, \widetilde{E}, \widetilde{w}) \leftarrow (V, \emptyset, 0)$
3:    **for all** $e \in E$ **do**
4:        $R^{\max}(e) \leftarrow \max_{f \in F_e} R_G(f)$
5:        $p(e) \leftarrow \min(1, w(e) \cdot R^{\max}(e) \cdot \lambda)$
6:        With probability $p(e)$, $\widetilde{E} \leftarrow E \cup \{e\}$ and $\widetilde{w}(e) \leftarrow w(e)/p(e)$
7:    **return** $\widetilde{H}$

In the rest of this section, we show the correctness of this approach. We first bound the size of the hypergraph output by Algorithm 3. We then prove that the output $\widetilde{H}$ is indeed an ϵ-spectral sparsifier of H—this is the technical core of the section.

Lemma 6.1. *Let $H = (V, E, w)$ be a weighted hypergraph and let $G = (V, F, z)$ be its $1/n^2$-approximate γ-balanced weight assignment for $\gamma \geq 4$. Then, Algorithm 3 returns a hypergraph of expected size $\mathbb{E}(|\widetilde{E}|) \leq 2\lambda\gamma n$.*

Proof. Each hyperedge e contributes $p(e) \leq w(e) \cdot R^{\max}(e) \cdot \lambda$ to $\mathbb{E}(|\widetilde{E}|)$, and it thus suffices to bound

$$\sum_{e \in E} w(e) \cdot R^{\max}(e) \leq 2\gamma \cdot n. \tag{11}$$

We proceed to prove this inequality. For each $e \in E$, let us partition F_e into two groups, $F_e^{(1)} = \{f \in F_e \mid \gamma \cdot R_G(f) \geq R^{\max}(e)\}$ and the remaining edges $F_e^{(2)} = F_e \setminus F_e^{(1)}$. By Definition 5.9, these remaining edges $f \in F_e^{(2)}$ satisfy $z(f) \leq w(e)/n^2$. Then, by Fact 2.9, we have

$$n - 1 = \sum_{f \in F} z(f) \cdot R_G(f) = \sum_{e \in E} \sum_{f \in F_e} z(f) \cdot R_G(f) \geq \sum_{e \in E} \sum_{f \in F_e^{(1)}} z(f) \cdot \frac{1}{\gamma} \cdot R^{\max}(e),$$

where the second equality is due to the fact that F is the multiset $\bigcup_{e \in E} F_e$ by Definition 5.1. For the inner summation, we can bound

$$\sum_{f \in F_e^{(1)}} z(f) \geq \sum_{f \in F_e} z(f) - \sum_{f \in F_e^{(2)}} \frac{w(e)}{n^2} \geq w(e) - \frac{1}{2} \cdot w(e) = \frac{1}{2} \cdot w(e),$$

and altogether we obtain $\sum_{e \in E} w(e) \cdot R^{\max}(e) \leq 2\gamma \cdot (n - 1)$, which completes the proof. $\square$

Setting $\gamma = 4$, for example, thus produces a linear-size output. We now prove that the output is indeed a spectral sparsifier of H.

Lemma 6.2. *Let $H = (V, E, w)$ be a weighted hypergraph and let $G = (V, F, z)$ be its $1/n^2$-approximate γ-balanced weight assignment for some* constant $\gamma \geq 4$. *Then executing Algorithm 3 on H, G, $1/n \leq \epsilon < 1$, and $\lambda = O(\log^3(n)/\epsilon^4)$, returns with probability at least $1 - O(\log(n)/n)$ an ϵ-spectral sparsifier of H.*

Proof. We proceed similarly to our proof of Theorem 4.1 in Section 4. By Definition 2.4, we must prove that for every $x \in \mathbb{R}^V$,

$$\left| Q_H(x) - Q_{\widetilde{H}}(x) \right| \leq \epsilon \cdot Q_H(x), \tag{12}$$

Since Equation (12) is invariant to scaling, we may assume without loss of generality that $x^\top L_G x = 1$, and we denote the set of such vectors by $S^G \subseteq \mathbb{R}^V$. Notice that S^G is defined with respect to L_G and not Q_H, however it implies that $Q_H(x) \geq 1$ because for all $x \in S^G$,

$$Q_H(x) = \sum_{e \in E} w(e) \cdot \max_{u^*, v^* \in e} (x_{u^*} - x_{v^*})^2 = \sum_{e \in E} \sum_{f \in F_e} z(f) \cdot \max_{u^*, v^* \in e} (x_{u^*} - x_{v^*})^2 \tag{13}$$

$$\geq \sum_{e \in E} \sum_{f = (u,v) \in F_e} z(f) \cdot (x_u - x_v)^2 = \sum_{f = (u,v) \in F} z(f) \cdot (x_u - x_v)^2 = x^\top L_G x = 1.$$

For any $E' \subseteq E$, we denote the quadratic form $Q(x)$ restricted only to the hyperedges E' in H by $Q_{E'}(x)$, and similarly in the hypergraph $\widetilde{H}$ by $\widetilde{Q}_{E'}(x)$. If $E' = E$, we omit the subscript. It is clear by the construction of $\widetilde{H}$ that

$$\mathbb{E}\left(\widetilde{Q}(x) \right) = Q(x).$$

Therefore, for any specific vector x, Equation (12) holds by Chernoff bound (Theorem 2.10). In order to prove it simultaneously for all $x \in S^G$, we again use progressively finer and finer roundings of x, as guaranteed by Lemma 4.2.

Indeed, fix a sequence of the rounding functions φ_i guaranteed by Lemma 4.2 for $i = 1, \ldots, I := \log_2(14\gamma n/\epsilon) \leq 3 \log n$ (since γ is a fixed constant and n is sufficiently large), and denote the sequence of rounded vectors for $x \in S^G$ by $x^{(i)} = \varphi_i(x)$. We define for each $x^{(i)}$ the set of hyperedges

$$E'_i := \left\{ e \in E \;\middle|\; \max_{u,v \in e} \left(x_u^{(i)} - x_v^{(i)} \right)^2 \geq 2^{-i} \cdot R^{\max}(e) \right\},$$

where by definition $R^{\max}(e) = \max_{f \in F_e} R_G(f)$. This set E'_i is designed such that the rounding φ_i will conserve (to within a small multiplicative error) $Q_{\{e\}}(x)$ for any $e \in E'_i$ (we will see a proof of this fact later on). We can now define a partition of E based on the sets E'_i. First, recall that $p(e) = \min(1, \lambda \cdot w(e) \cdot R^{\max}(e))$, and let the base case be $E_0 = E'_0 := \{e \in E \mid p(e) = 1\}$. Then for each $i = 1, \ldots, I$, let

$$E_i := E'_i \setminus \bigcup_{j=0}^{i-1} E'_j.$$

Finally, let $E_{I+1} := E \setminus \bigcup_{i=0}^I E'_i$.

We begin with two useful claims that will help us bridge the gap between the ordinary graph and the hypergraph settings. First, we extend the second guarantee of Lemma 4.2 to hyperedges.

Claim 6.3. *For all $x \in S^G$ and $e \in E$ such that*

$$\max \left(\max_{u,v \in e} (x_u - x_v)^2, \max_{u,v \in e} (x_u^{(i)} - x_v^{(i)})^2 \right) \geq 2^{-i} \cdot R^{\max}(e),$$

we have

$$\max_{u,v \in e} (x_u - x_v)^2 \in \left(1 \pm \frac{\epsilon}{7} \right) \cdot \max_{u,v \in e} (x_u^{(i)} - x_v^{(i)})^2. \tag{14}$$

Proof. We focus on the case when $\max_{u,v \in e}(x_u^{(i)} - x_v^{(i)})^2 \geq 2^{-i} \cdot R^{\max}(e)$. The case when $\max_{u,v \in e}(x_u - x_v)^2 \geq 2^{-i} \cdot R^{\max}(e)$ follows by an identical argument.

22

For one direction, let the pair $u^*, v^* \in e$ maximize $(x_u^{(i)} - x_v^{(i)})^2$. Now

$$(x_{u^*}^{(i)} - x_{v^*}^{(i)})^2 = \max_{u,v \in E}(x_u^{(i)} - x_v^{(i)})^2 \geq 2^{-i} \cdot R^{\max}(e) = 2^{-i} \cdot \max_{f \in F_e} R_G(f) \geq 2^{-i} \cdot R_G(u^*, v^*),$$

hence the second guarantee of Lemma 4.2 holds for (u^*, v^*), and consequently

$$(x_{u^*}^{(i)} - x_{v^*}^{(i)})^2 \leq \left(1 - \frac{\epsilon}{7}\right)^{-1} \cdot (x_{u^*} - x_{v^*})^2 \leq \left(1 - \frac{\epsilon}{7}\right)^{-1} \cdot \max_{u,v \in e}(x_u - x_v)^2.$$

The other direction follows by a similar argument, but is slightly more complicated. The asymmetry is due to our assumption (that $\max_{u,v \in e}(x_u^{(i)} - x_v^{(i)}) \geq 2^{-i} \cdot R^{\max}(e)$) being in terms of $x^{(i)}$ instead of x.

Let u^*, v^* be a vertex pair $u, v \in e$ that maximizes $(x_u - x_v)^2$. Here we distinguish two cases: If $(x_{u^*} - x_{v^*})^2 < 2^{-i} \cdot R^{\max}(e)$, we are immediately done, since

$$\max_{u,v \in e}(x_u - x_v)^2 = (x_{u^*} - x_{v^*})^2 \leq 2^{-i} \cdot R^{\max}(e) \leq \max_{u,v \in e}(x_u^{(i)} - x_v^{(i)})^2.$$

On the other hand, if $(x_{u^*} - x_{v^*})^2 \geq 2^{-i} \cdot R^{\max}(e)$, we proceed identically to the first half of the proof:

$$(x_{u^*} - x_{v^*})^2 \geq 2^{-i} \cdot R^{\max}(e) = 2^{-i} \cdot \max_{f \in F_e} R_G(f) \geq 2^{-i} \cdot R_G(u^*, v^*).$$

Therefore, the second guarantee of Lemma 4.2 holds for (u^*, v^*) in G, and in particular

$$(x_{u^*} - x_{v^*})^2 \leq \left(1 + \frac{\epsilon}{7}\right) \cdot (x_{u^*}^{(i)} - x_{v^*}^{(i)})^2 \leq \left(1 + \frac{\epsilon}{7}\right) \cdot \max_{u,v \in e}(x_u^{(i)} - x_v^{(i)})^2. \qquad \square$$

We thus obtain an analogue to Corollary 4.3 from Section 4.

Corollary 6.4. *For every edge $e \in E_i$,*

$$Q_{\{e\}}(x) \in \left(1 \pm \frac{\epsilon}{7}\right) \cdot Q_{\{e\}}(x^{(i)}).$$

The same holds also for $\widetilde{Q}_{\{e\}}$ (because it is a multiple of $Q_{\{e\}}$).

Proof. We have that $e \in E_i \subseteq E_i'$ and hence $\max_{u,v \in e}(x_u^{(i)} - x_v^{(i)})^2 \geq 2^{-i} \cdot R^{\max}(e)$. We can therefore apply Claim 6.3 and scale up by $w(e)$ to get the desired result. $\qquad \square$

Next, we prove an analogue to Claim 4.4 from Section 4.

Claim 6.5. *For all $i \in [I]$ and $e \in E_i$, we have*

$$\max_{u,v \in e}(x_u^{(i)} - x_v^{(i)})^2 \leq 3 \cdot 2^{-i} \cdot R^{\max}(e).$$

Proof. The proof proceeds nearly identically to that of Claim 4.4.

The condition of Claim 6.3 holds for $e \in E_i$ and hence we have $\max_{u,v \in e}(x_u^{(i)} - x_v^{(i)})^2 \leq (1 - \epsilon/7)^{-1} \cdot \max_{u,v \in e}(x_u - x_v)^2$.

Consider first the case $i = 1$. Let $u^*, v^* \in e$ maximize $(x_u - x_v)^2$. Then by Fact 2.6 and since $x \in S^G$, we have $\max_{u,v \in e}(x_u - x_v)^2 = (x_{u^*} - x_{v^*})^2 \leq R_G(u^*, v^*) \cdot x^\top L_G x = R_G(u^*, v^*)$, and we indeed get

$$\max_{u,v \in e}(x_u^{(i)} - x_v^{(i)})^2 \leq \max_{u,v \in e}(x_u - x_v)^2 \cdot \left(1 - \frac{\epsilon}{7}\right)^{-1}$$

$$\leq R_G(u^*, v^*) \cdot \left(1 - \frac{\epsilon}{7}\right)^{-1}$$

$$\leq R^{\max}(e) \cdot \left(1 - \frac{\epsilon}{7}\right)^{-1}$$

$$\leq 3 \cdot 2^{-1} \cdot R^{\max}(e).$$

Now consider $i > 1$, and suppose towards contradiction that $\max_{u,v \in e}(x_u^{(i)} - x_v^{(i)})^2 > 3 \cdot 2^{-i} \cdot R^{\max}(e)$. Notice that since $\max_{u,v \in e}(x_u - x_v)^2 \geq 2^{-i+1} \cdot R^{\max}(e)$, Claim 6.3 still applies to e with respect to φ_{i-1}. Thus

$$\max_{u,v \in e}(x_u^{(i-1)} - x_v^{(i-1)})^2 \geq \max_{u,v \in e}(x_u - x_v)^2 \cdot \left(1 + \frac{\epsilon}{7}\right)^{-1}$$

$$\geq \max_{u,v \in e}(x_u^{(i)} - x_v^{(i)})^2 \cdot \left(1 - \frac{\epsilon}{7}\right) \cdot \left(1 + \frac{\epsilon}{7}\right)^{-1}$$

$$\geq 2^{-i+1} \cdot R^{\max}(e).$$

This implies that $e \in E'_{i-1}$, which contradicts the assumption $e \in E_i$. $\qquad\square$

We consider each group of hyperedges E_i separately, and prove that the quadratic form is well approximated on this subset with high probability, that is

$$\widetilde{Q}_{E_i}(x) \approx Q_{E_i}(x).$$

Hyperedges in E_0. By definition, for all $e \in E_0$, we have $p(e) = 1$. Therefore, $\widetilde{Q}_{E_0}(x) = Q_{E_0}(x)$ deterministically.

Hyperedges in E_i for $i \in [I]$. Similarly to the case of ordinary graphs in Section 4, we want to consider $x^{(i)}$ instead of x. We may do this due to Corollary 6.4. Therefore, we can focus on proving that $Q_{E_i}(x^{(i)}) \approx \widetilde{Q}_{E_i}(x^{(i)})$.

Consider a specific $i \in [I]$. We wish to prove that with high probability, simultaneously for all values of $x^{(i)}$ and $E^{(i)}$,

$$\left|\widetilde{Q}_{E_i}(x^{(i)}) - Q_{E_i}(x^{(i)})\right| \leq \frac{\epsilon \cdot Q(x)}{7I}. \tag{15}$$

Since (by the first guarantee of Lemma 4.2) there are only finitely many values $(x^{(i)}, E_i)$, we can prove that Equation (15) holds with high probability individually for each such pair, and then use a union bound over all pairs.

The former can be done using Chernoff bounds (Theorem 2.10); we begin with this. Let us fix $x^{(i)}$ and E_i. Recall that

$$\widetilde{Q}_{E_i}(x^{(i)}) = \sum_{e \in E_i} \widetilde{w}(e) \cdot \max_{u,v \in e}(x_u^{(i)} - x_v^{(i)})^2$$

24

is the sum of independent random variables with expectation $\mathbb{E}(\widetilde{Q}_{E_i}(x^{(i)})) = Q_{E_i}(x^{(i)})$ because $\mathbb{E}(\widetilde{w}(e)) = w(e)$ by definition. To apply Chernoff bounds (Theorem 2.10), we need to set the variables a, δ and μ. We set $\delta := \epsilon/(14I)$, and using Corollary 6.4, we can bound

$$Q_{E_i}(x^{(i)}) \le \left(1 - \frac{\epsilon}{7}\right)^{-1} \cdot Q_{E_i}(x) \le 2Q(x) =: \mu.$$

(This is true for an arbitrary preimage $x \in \varphi_i^{-1}(x^{(i)})$.) We also need to set a as an upper bound on the largest possible value of any variable of the form $\widetilde{w}(e) \cdot \max_{u,v \in e}(x_u^{(i)} - x_v^{(i)})^2$ for $e \in E_i$. Such a variable takes its maximum value when e is sampled to the sparsifier $\widetilde{H}$, in which case $\widetilde{w}(e) = w(e)/p(e) = w(e)/(\lambda \cdot w(e) \cdot R^{\max}(e)) = 1/(\lambda \cdot R^{\max}(e))$, since $e \notin E_0$. Therefore, the random variable in question is upper bounded, using Claim 6.5, by

$$\max_{e \in E_i} \frac{\max_{u,v \in e}(x_u^{(i)} - x_v^{(i)})^2}{\lambda \cdot R^{\max}(e)} \le \frac{3 \cdot 2^{-i}}{\lambda} =: a.$$

Finally, Theorem 2.10 implies

$$\mathbb{P}\left(\left|Q_{E_i}(x^{(i)} - \widetilde{Q}_{E_i}(x^{(i)}))\right| \ge \frac{\epsilon \cdot Q(x)}{7I}\right) \le 2\exp\left(-\frac{\delta^2 \mu}{3a}\right)$$

$$= 2\exp\left(-\frac{\frac{\epsilon^2}{196I^2} \cdot Q(x)}{3 \cdot 3 \cdot 2^{-i}/\lambda}\right)$$

$$\le 2\exp\left(-\frac{\epsilon^2 \cdot 2^i \cdot \lambda}{10000 \log^2(n)}\right)$$

$$= 2\exp\left(-\frac{2000C \log(n) \cdot 2^i}{\epsilon^2}\right),$$

where the last step is by setting $\lambda = 2 \cdot 10^7 \cdot C \log^3(n)/\epsilon^4$.

We now turn to applying a union bound over all possible values of $(x^{(i)}, E_i)$. Much like F_i in the proof of Theorem 4.1, E_i depends on all E_j' for $j \le i$, which in turn depend on all $x^{(j)}$ for $j \le i$. Since $x^{(j)} = \varphi_j(x)$, by the first guarantee of Lemma 4.2, there are at most $\exp(800C \log(n) \cdot 2^j/\epsilon^2)$ possible vectors $x^{(j)}$ (where C is the absolute constant from Theorem 4.5). Therefore, the number of possible pairs $(x^{(i)}, E_i)$ is at most

$$\prod_{j=1}^{i} \exp\left(\frac{800C \log(n) \cdot 2^j}{\epsilon^2}\right) = \exp\left(\sum_{j=1}^{i} \frac{800C \log(n) \cdot 2^j}{\epsilon^2}\right) \le \exp\left(\frac{1600C \log(n) \cdot 2^i}{\epsilon^2}\right).$$

Putting together the Chernoff bound and the union bound, we get that Equation (15) holds simultaneously for all values of $(x^{(i)}, E_i)$ except with probability at most

$$\exp\left(\frac{1600C \log(n) \cdot 2^i}{\epsilon^2}\right) \cdot 2\exp\left(-\frac{2000C \log(n) \cdot 2^i}{\epsilon^2}\right) = 2\exp\left(-\frac{400C \log(n) \cdot 2^i}{\epsilon^2}\right) \le \frac{1}{n}.$$

Hyperedges in E_{I+1}. Recall that $I = \log_2(14\gamma \cdot n/\epsilon)$. First we show that for any hyperedge $e \in E_{I+1}$, we have that $\max_{u,v \in e}(x_u - x_v)^2 \le \epsilon \cdot R^{\max}(e)/(12\gamma \cdot n)$. Indeed, suppose for contradiction that this is not the case, and let $u^*, v^* \in e$ be a vertex pair that maximizes $(x_u - x_v)^2$. Then, by the second guarantee of Lemma 4.2, we have that $(x_{u^*}^{(I)} - x_{v^*}^{(I)})^2 \ge (x_{u^*} - x_{v^*})^2 \cdot (1 - \epsilon/7) \ge$

$\epsilon \cdot R^{\max}(e) \cdot (1 - \epsilon/7)/(12\gamma \cdot n) \geq \epsilon \cdot R^{\max}(e)/(14\gamma \cdot n)$. Therefore, $e \in E_I'$, which contradicts $e \in E_{I+1}$.

We show that $|\widetilde{Q}_{E_{I+1}} - Q_{E_{I+1}}|$ is small by showing that each of the two terms is individually small. First,

$$Q_{E_{I+1}}(x) = \sum_{e \in E_{I+1}} w(e) \cdot \max_{u,v \in e}(x_u - x_v)^2 \leq \sum_{e \in E_{I+1}} w(e) \cdot \frac{\epsilon \cdot R^{\max}(e)}{12\gamma \cdot n} \leq \frac{\epsilon}{12\gamma \cdot n} \cdot \sum_{e \in E} w(e) \cdot R^{\max}(e) \leq \frac{\epsilon}{6},$$

where the last inequality uses Equation (11). Second, we start similarly,

$$\widetilde{Q}_{E_{I+1}}(x) = \sum_{e \in E_{I+1}} \widetilde{w}(e) \cdot \max_{u,v \in e}(x_u - x_v)^2 \leq \sum_{e \in E_{I+1}} \widetilde{w}(e) \cdot \frac{\epsilon \cdot R^{\max}(e)}{12\gamma \cdot n} \leq \frac{\epsilon}{12\gamma \cdot n} \cdot \sum_{e \in E} \widetilde{w}(e) \cdot R^{\max}(e),$$

and ideally we would like to show that $\sum \widetilde{w}(e) \cdot R^{\max}(e) \leq 4\gamma \cdot n$. This is a random event, independent of the choice of x, whose probability we can bound using the Chernoff bound (Theorem 2.10). Recall that $E_0 = \{e \in E \mid p(e) = 1\}$, and denote its complement by $\overline{E}_0 := \{e \in E \mid p(e) = \lambda \cdot w(e) \cdot R^{\max}(e)\}$. Since $\mathbb{E}(\sum \widetilde{w}(e) \cdot R^{\max}(e)) = \sum w(e) \cdot R^{\max}(e) \leq 2\gamma \cdot n$ (using Equation (11)) and $\sum_{e \in E_0} \widetilde{w}(e) \cdot R^{\max}(e)$ is deterministic by definition of E_0, we have that

$$\mathbb{P}\left(\sum_{e \in E} \widetilde{w}(e) \cdot R^{\max}(e) \geq 4\gamma \cdot n\right) \leq \mathbb{P}\left(\left|\sum_{e \in E} \widetilde{w}(e) \cdot R^{\max}(e) - \mathbb{E}\left(\sum_{e \in E} \widetilde{w}(e) \cdot R^{\max}(e)\right)\right| \geq 2\gamma \cdot n\right)$$

$$= \mathbb{P}\left(\left|\sum_{e \in \overline{E}_0} \widetilde{w}(e) \cdot R^{\max}(e) - \mathbb{E}\left(\sum_{e \in \overline{E}_0} \widetilde{w}(e) \cdot R^{\max}(e)\right)\right| \geq 2\gamma \cdot n\right).$$

We bound this by applying Theorem 2.10 for the independent random variables $\widetilde{w}(e) \cdot R^{\max}(e)$ where $e \in \overline{E}_0$. The maximum value of such a variable occurs when e is sampled, in which case it is exactly $\widetilde{w}(e) \cdot R^{\max}(e) = R^{\max}(e) \cdot w(e)/p(e) = 1/\lambda \leq 1 =: a$. Setting $\delta := 1$ and $\mu := 2\gamma \cdot n$ (we may do this due to Equation (11)), we get

$$\mathbb{P}\left(\sum_{e \in E} \widetilde{w}(e) \cdot R^{\max}(e) \geq 4\gamma \cdot n\right) \leq 2 \exp\left(-\frac{2\gamma \cdot n}{3}\right) \leq \frac{1}{n}.$$

In conclusion, with probability at least $1 - O(1/n)$,

$$\left|\widetilde{Q}_{E_{I+1}}(x) - Q_{E_{I+1}}(x)\right| \leq \frac{\epsilon}{2}. \tag{16}$$

Putting everything together. The final part of the proof proceeds identically to that of Theorem 4.1. We have seen that Equation (15) holds simultaneously for all i and $(x^{(i)}, E_i)$, as well as Equation (16) holds with probability at least $1 - O(\log(n)/n)$. Conditioning on these events, we can deduce Equation (12), thus concluding the proof of Lemma 6.2.

For completeness we repeat the derivation:

$$\left|Q_{E_i}(x) - \widetilde{Q}_{E_i}(x)\right| \leq \left|Q_{E_i}(x) - Q_{E_i}(x^{(i)})\right| + \left|Q_{E_i}(x^{(i)}) - \widetilde{Q}_{E_i}(x^{(i)})\right| + \left|\widetilde{Q}_{E_i}(x^{(i)}) - \widetilde{Q}_{E_i}(x)\right|$$

$$\leq \frac{\epsilon}{7} \cdot Q_{E_i}(x^{(i)}) + \left|Q_{E_i}(x^{(i)}) - \widetilde{Q}_{E_i}(x^{(i)})\right| + \frac{\epsilon}{7} \cdot \widetilde{Q}_{E_i}(x^{(i)}) \quad \text{(by Corollary 6.4)}$$

$$\leq \frac{2\epsilon}{7} \cdot Q_{E_i}(x^{(i)}) + \left(1 + \frac{\epsilon}{7}\right) \cdot \left|Q_{E_i}(x^{(i)}) - \widetilde{Q}_{E_i}(x^{(i)})\right|$$

$$\leq \frac{2\epsilon}{7} \cdot \left(1 - \frac{\epsilon}{7}\right)^{-1} \cdot Q_{E_i}(x) + \left(1 + \frac{\epsilon}{7}\right) \cdot \frac{\epsilon \cdot Q(x)}{7I}$$

$$\text{(by Corollary 6.4 and Equation (15))}$$

$$\leq \frac{\epsilon}{3} \cdot Q_{E_i}(x) + \frac{\epsilon \cdot Q(x)}{6I}.$$

Therefore, we have

$$\left| Q(x) - \widetilde{Q}(x) \right| \leq \left| Q_{E_0}(x) - \widetilde{Q}_{E_0}(x) \right| + \sum_{i=1}^{I} \left| Q_{E_i}(x) - \widetilde{Q}_{E_i}(x) \right| + \left| Q_{E_{I+1}}(x) - \widetilde{Q}_{E_{I+1}}(x) \right|$$

$$\leq 0 + \sum_{i=1}^{I} \left(\frac{\epsilon}{3} \cdot Q_{E_i}(x) + \frac{\epsilon \cdot Q(x)}{6I} \right) + \frac{\epsilon}{2} \qquad \text{(by Equation (16))}$$

$$\leq \frac{\epsilon}{3} \cdot Q(x) + \frac{\epsilon}{6} \cdot Q(x) + \frac{\epsilon}{2}$$

$$\leq \epsilon \cdot Q(x), \qquad \text{(by Equation (13))}$$

as claimed. □

Combining Theorem 5.10 and Lemmas 6.1 and 6.2, we get the first part of Theorem 1.

7   Nearly Optimal Speed-Up

In the previous section, we have proved the existence of nearly linear sized spectral sparsifiers for hypergraphs. We have also provided an method for constructing such sparsifiers: we construct an approximate balanced weight assignment of the input hypergraph using Algorithm 2, and then construct a sparsifier using Algorithm 3. However, the running time of Algorithm 2 on an unweighted hypergraph is $m \cdot \text{poly}(n)$, which is large; in this section we improve this to $\widetilde{O}(mr) + \text{poly}(n)$, that is, we prove the second part of Theorem 1. As we mentioned in the introduction, with a small modification, this leads to an algorithm with time complexity $\widetilde{O}(\sum_{e \in E} |e| + \text{poly}(n))$ that constructs an ϵ-spectral-sparsifier of nearly linear size. This running time is optimal to within polylogarithmic factors in n, unless the size of the input hypergraph is polynomially small in n.

Our algorithm consists of two steps. First we apply the algorithm of [BST19], which—with small modifications—can be shown to run in $\widetilde{O}(mr) + \text{poly}(n)$ time, but which produces a larger spectral sparsifier. We then aim to sparsify the resulting weighted hypergraph using our methods. Unfortunately, even though the resulting hypergraph has only polynomially many hyperedges (in n), the range of edge weights may still be exponential, meaning that Algorithm 2 is not efficient for finding an approximate balanced weight assignment on it (recall Theorem 5.10). We propose a variation of Algorithm 2 adapted for this setting which runs in polynomial time.

7.1   Fast Algorithm for Constructing Polynomial-Sized Sparsifiers

In this section, we recall and slightly modify the algorithm of [BST19] for producing polynomial-sized spectral sparsifiers for hypergraphs.

Definition 7.1. *For a weighted hypergraph $H = (V, E)$, let $G(H)$ denote the "associated graph" of H, which is defined as follows: Replace each hyperedge e of E with a clique of uniform weight $w(e)$ on the support of e. (Note that this may produce parallel edges.)*

Theorem 7.2 ([BST19]). *Let $H = (V, E, w)$ be a hypergraph where all hyperedges have size between $r/2$ and r. Then, for some absolute constant c, the following process produces an ϵ-spectral sparsifier for H with probability at least $1 - 1/n$: Let $G(H)$ be the associated graph of H. For each hyperedge $e \in E$, let*

$$R^{\max}(e) = \max_{u,v \in e} R_{G(H)}(u, v).$$

Sample each hyperedge e independently with some probability

$$p(e) \geq \min\left(1, \frac{w(e) \cdot R^{\max}(e) \cdot r^4 \log n}{c\epsilon^2}\right), \tag{17}$$

and if sampled give it weight $\widetilde{w}(e) = w(e)/p(e)$.

Remark 7.3. *In fact, in [BST19], the result is stated slightly less generally, for unweighted hypergraphs, and without allowing oversampling (that is $p(e)$ is set exactly to the right hand side in Equation 17, instead of being lower bounded by it). However, this version holds by an identical proof.*

The trivial implementation of this takes time $\Omega(mr^2)$ in general for two different reasons: First, it takes $\Omega(mr^2)$ time to replace each of the m hyperedges with a clique of size r^2. Second, it takes $\Omega(mr^2)$ time to find $\max_{u,v \in e} R_{G(H)}(u, v)$ for all m hyperedges. However, with some simple tricks in the implementation both bottlenecks can be avoided to reduce the running time to $\widetilde{O}(mr + \text{poly}(n))$.

To achieve this, we first replace cliques in the associated graph of the input hypergraph with sparse graphs guaranteed by the following fact:

Fact 7.4. *It follows by Theorem 4.1 that for every r, there exists a (weighted) graph G_r^* with r vertices and $\widetilde{O}(r)$ edges such that G_r^* is a $1/2$-spectral sparsifier to the r-clique.*

Second, to calculate $R^{\max}(e)$ approximately, we do not take the maximum over all pairs of vertices in e, but only over (u_0, v) for all $v \in e$ but for some fixed u_0. Since effective resistance is a metric (see Fact 2.7), this provides a 2-approximation to $R^{\max}(e)$ by triangle inequality.

Algorithm 4 Fast algorithm for computing a polynomial-sized spectral sparsifier for an approximately uniform hypergraph.

1: **procedure** UNIFORMSPARSIFICATION($H = (V, E, w), r, \epsilon$)
2:     $G = (V, F, z) \leftarrow (V, \emptyset, 0)$
3:     **for all** $e \in E$ **do**
4:         Add a copy of $G_{|e|}^*$ to G, supported on e, with weights scaled up by $w(e)$
5:     Calculate $R_G(u, v)$ for all $u, v \in V$
6:     $\widetilde{H} = (V, \widetilde{E}, \widetilde{w}) \leftarrow (V, \emptyset, 0)$
7:     **for all** $e \in E$ **do**
8:         $u_0 \leftarrow$ an arbitrary vertex in e
9:         $\widetilde{R}^{\max}(e) \leftarrow \max_{v \in e} R_G(u_0, v)$
10:       $p(e) \leftarrow \min\left(1, \frac{4w(e) \cdot \widetilde{R}^{\max}(e) \cdot r^4 \log n}{c\epsilon^2}\right)$
11:       Add e to $\widetilde{E}$ with probability $p(e)$ with weight $\widetilde{w}(e) \leftarrow w(e)/p(e)$
12:     **return** $\widetilde{H}$

Lemma 7.5. *If the input hypergraph $H = (V, E, w)$ only has hyperedges of size between $r/2$ and r, then Algorithm 4 runs in $\widetilde{O}(mr) + \text{poly}(n)$ time, returning an ϵ-spectral sparsifier to H with*

probability at least $1 - 1/n$. Furthermore, the output has at most $4n^5 \log n/(c\epsilon^2)$ hyperedges in expectation, where c is the absolute constant from Theorem 7.2.

Proof. It takes $\widetilde{O}(mr)$ time to construct the graph G (here Line 4 takes $\widetilde{O}(r)$ time) and its Laplacian. All pairs effective resistances can then be calculated in poly (n) time and stored in a table, resulting in an $O(r)$ time bound for the calculation of $\widetilde{R}^{\max}(e)$ in Line 9.

To show that the output is an ϵ-spectral sparsifier of H with high probability, it suffices to verify Equation 17 of Theorem 7.2, ie. that $p(e)$ is always at least

$$w(e) \cdot \max_{u,v \in e} R_{G(H)}(u,v) \cdot \frac{r^4 \log n}{c\epsilon^2}.$$

For this, it suffices to show that $\widetilde{R}^{\max}(e)$ (as defined in Line 9 of Algorithm 4) is at least $\max_{u,v \in e}$ $R_{G(H)}(u,v)/4$. Since $w(e) \cdot G^*_{|e|}$ is a 1/2-spectral sparsifier of the clique on e (of uniform edge weight $w(e)$), it follows by the additivity of Laplacians that G from Algorithm 4 is a 1/2-spectral sparsifier of $G(H)$ from Definition 7.1. Therefore, $R_G(u,v) \geq R_{G(H)}(u,v)/2$ for all $u,v \in V$ (recall Definition 2.5). Finally, since R_G is a metric on V (by Fact 2.7),

$$\max_{v \in e} R_G(u_0,v) \geq \max_{u,v \in e} R_G(u,v)/2.$$

Indeed, if (u^*,v^*) maximizes $R_G(u,v)$, then by triangle inequality $R_G(u^*,v^*) \leq R_G(u^*,u_0) + R_G(u_0,v^*)$; one of the latter two must be at least $R_G(u^*,v^*)/2$.

This concludes the proof of correctness; we must finally prove the upper bound on the expected size of the output $\widetilde{H}$. Since $r^4 \leq n^4$, it suffices to show that $\sum_{e \in E} w(e) \cdot \widetilde{R}^{\max}(e) \leq n$. First note that

$$\sum_{e \in E} w(e) \cdot \widetilde{R}^{\max}(e) \leq \sum_{e \in E} w(e) \cdot \max_{u,v \in e} R_G(u,v) \leq 2 \sum_{e \in E} w(e) \cdot R_{G(H)}(u,v),$$

since G is a 1/2-spectral sparsifier of $G(H)$. Then, since the weight of (u,v) in $G(H)$ is exactly $w(e)$ by definition, we have that

$$\sum_{e \in E} w(e) \cdot \widetilde{R}^{\max}(e) \leq \sum_{f \in F(H)} z_{G(H)}(f) \cdot R_{G(H)}(f) = n - 1,$$

by Fact 2.9, where $F(H)$ denotes the edge-set of $G(H)$ and $z_{G(H)}$ denotes the weight function of $G(H)$. $\square$

Algorithm 4 only works under the constraint that the input hypergraph is approximately uniform – that is, the sizes of hyperedges all fall into the range $[r/2, r]$. This is easily circumvented, however: Given an arbitrary input hypergraph, one can simply partition the hyperedges into a logarithmic number of parts based on cardinality. We then sparsify the parts, and combine the resulting sparsifiers, losing only a $\log n$ factor in size.

We formalize this in the following algorithm and corollary.

Algorithm 5 Fast algorithm for computing a polynomial-sized spectral sparsifier for an arbitrary hypergraph.

1: **procedure** POLYNOMIALSIZESPARSIFICATION($H = (V,E,w), \epsilon$)
2:      **for** i from 1 to $\log n$ **do**
3:          $E_i \leftarrow \{e \in E \, | \, |e| \in [2^i, 2^{i+1})\}$
4:          $H_i \leftarrow (V, E_i, w)$
5:          $\widetilde{H}_i \leftarrow$ UNIFORMSPARSIFICATION($H_i, 2^{i+1}, \epsilon$) $\triangleright$ Algorithm 4
6:      **return** $\widetilde{H} \leftarrow \cup_{i=1}^{\log n} \widetilde{H}_i$

Here the final line means that we take all hyperedges (with associated weights) from all of $\widetilde{H}_1, \ldots, \widetilde{H}_{\log n}$.

Corollary 7.6. *Algorithm 5 runs in $\widetilde{O}(mr) + \text{poly}(n)$ time, returning an ϵ-spectral sprasifier of the input H with probability at least $1 - \log n/n$. Furthermore, the output has at most $4n^5 \log^2(n)/(c\epsilon^2)$ hyperedges in expectation, where c is the absolute constant from Theorem 7.2.*

7.2 Even Faster Construction for γ-Balanced Weight Assignments

It is difficult to get a stronger bound on the number of rounds of Algorithm 2, than that of Theorem 5.10, at least in its full generality. Instead, here we define a specific class of hypergraphs for which a better algorithm exists.

Definition 7.7. *A weighted hypergraph $H = (V, E, w)$ is called (α, β)-separated for parameters $\alpha \geq 1$ and $\beta \geq 1$ if the hyperedge set E is partitioned into $E_1, \ldots, E_\ell$, satisfying the two requirements:*

$$\text{for all } i \in \{1, 2, \ldots, \ell\}, \qquad\qquad \max_{e \in E_i} w(e) \leq \alpha \cdot \min_{e \in E_i} w(e),$$

$$\text{for all } i, j \in \{1, 2, \ldots, \ell\}, i < j, \qquad\qquad \min_{e \in E_i} w(e) \geq \beta \cdot \max_{e \in E_j} w(e).$$

Our next algorithm exploits the structure of separated hypergraphs in order to more efficiently construct approximate balanced weight assignments on them. Intuitively, one can think of the different weight classes in separated hypergraphs as only weakly interacting, which is the source of our speedup. In particular, it is important to note that we will get a speedup for hypergraphs where the total number of hyperedges is small in comparison to the amount of separation between the weight classes (i.e., hypergraphs that are produced by the sparsification procedure from the previous section)—this is what ultimately ensures that the different weight classes only interact in a limited manner.

Algorithm 6 Computing an η-approximate γ-balanced weight assignment for an (α, β)-separated hypergraph.

1: **procedure** SEPARATEDAPPROXBALANCING($H = (V, E, w), (E_i)_{i=1}^\ell, \alpha, \beta, \gamma, \eta$)
2:      For all $e \in E$ and for all $f \in F_e$ initialize $z(f) \leftarrow w(e)/\binom{|e|}{2}$
3:      $G \leftarrow (V, \bigcup F_e, z)$
4:      **for** $i = 1, \ldots, \ell$ **do**
5:          **while** there exists $e \in E_i$ violating the conditions of η-approximate γ-balancedness **do**
6:              Select such $e \in E_i$ and $f, g \in F_e$, such that $R_G(f) > \gamma \cdot R_G(g)$ and $z(g) > \eta \cdot w(e)$
7:              $\lambda \leftarrow \min(z(g), 1/(2R_G(g)))$
8:              $z(f) \leftarrow z(f) + \lambda$
9:              $z(g) \leftarrow z(g) - \lambda$
10:     **return** G.

In fact, Algorithm 6 is very similar to Algorithm 2. However, it corrects discrepancies in heavier hyperedges first, and once a category of hyperedges (E_i) has been corrected, the algorithm never goes back to it, *not even if the approximate balancedness becomes violated.* For this reason, the resulting output, G, will not necessarily be γ-balanced. However, the structure of separated hypergraphs will allow us to bound the number of rounds using α, instead of the global aspect ratio of weights, which could be much larger.

The crucial property of separated hypergraphs (and their weight assignments) which we exploit is that the heavier edges have a much greater influence on the effective resistance of a vertex pair than the lighter edges. More specifically, for some $i \in [\ell]$ we can define the subgraph G_+ containing only edges in F_e for $e \in E_j$ where $j \leq i$. Then, when calculating the effective resistance $R_G(u, v)$—under certain circumstances—we can simply calculate $R_{G_+}(u, v)$ instead, ignoring the contribution of the remaining edges. We formalize this in the following lemma and its corollary.

Lemma 7.8. *Let $G_+ = (V, E_+, z_+)$ and $G_- = (V, E_-, z_-)$ be two weighted ordinary graphs on the same vertex set. Let the total weight of all edges in G_- be bounded by ζ. Let $G = G_+ \cup G_-$ be the union of the two graphs, that is $G = (V, E_+ \cup E_-, z_+ \cup z_-)$. Then, for any vertex pair $(u, v) \in V$ in the same connected component of G_+, we can bound the effective resistance $R_G(u, v)$ in terms of R_{G_+} as follows:*

$$\frac{1}{R_G(u, v)} \leq \frac{1}{R_{G_+}(u, v)} + \zeta.$$

Proof. To prove this inequality, we use the alternate definition of effective resistance from Fact 2.6. That is

$$R_{G_+}(u, v) = \max_{x \in \mathbb{R}^V} \frac{(x_u - x_v)^2}{x^\top L_{G_+} x}.$$

Let x^* maximize the above formula. We may assume without loss of generality that $x_u^* = 0$ and $x_v^* = 1$ since the formula is both shift and scale invariant. From this it follows that $x^{*\top} L_{G_+} x^* = 1/R_{G_+}(u, v)$. We can further assume without loss of generality that $x_a^* \in [0, 1]$ for all $a \in V$. Indeed, let the connected component of u and v in G_+ be $C \subseteq V$. Then for $a \in V \setminus C$, we may simply assume that x_a^* is always 0. To see that $x_a^* \in [0, 1]$ for all $a \in C$, suppose for contradiction that $\max_{a \in C} x_a^* > 1$. Let the highest value of x^* in C be $\mu_1 > 1$ and the second highest distinct value be μ_2. Then one can change the x^*-value of vertices from μ_1 to μ_2, thereby strictly increasing the value of $(x_u^* - x_v^*)^2 / x^{*\top} L_{G_+} x^*$. This is a contradiction, since x^* maximizes the formula by definition. An identical argument by contradiction rules out that $\min_{a \in C} x_a^* < 0$.

We can now upper bound $1/R_G(u, v)$ using the same alternate definition from Fact 2.6:

$$R_G(u, v) = \max_{x \in \mathbb{R}^V} \frac{(x_u - x_v)^2}{x^\top L_G x} \geq \frac{(x_u^* - x_v^*)^2}{x^{*\top} L_G x^*} = \frac{1}{1/R_{G_+}(u, v) + x^{*\top} L_{G_-} x^*}.$$

Now

$$x^{*\top} L_{G_-} x^* = \sum_{(a,b) \in E_-} z_-(a, b) \cdot (x_a^* - x_b^*)^2 \leq \sum_{(a,b) \in E_-} z_-(a, b) \leq \zeta.$$

Therefore

$$\frac{1}{R_G(u, v)} \leq \frac{1}{R_{G_+}(u, v)} + \zeta,$$

as claimed. $\square$

Corollary 7.9. *Let $G_+ = (V, E_+, z_+)$ and $G_- = (V, E_-, z_-)$ be two weighted ordinary graphs on the same vertex set. Let the total weight of all edges in G_- be bounded by ζ. Let $G = G_+ \cup G_-$ be the union of the two graphs, that is $G = (V, E_+ \cup E_-, z_+ \cup z_-)$. Then for any vertex pair $u, v \in V$, if $R_G(u, v) \leq 1/(5\zeta)$ then we can bound the effective resistance $R_G(u, v)$ in terms of $R_{G_+}(u, v)$ as follows:*

$$R_G(u, v) \geq \frac{4}{5} \cdot R_{G_+}(u, v).$$

Proof. We wish to apply Lemma 7.8, for which we must first show that u and v are in the same connected component in G_+. This is indeed the case: If u and v are in different connected components of G_+, we can use the alternate definition of effective resistance from Fact 2.6:

$$R_G(u,v) = \max_{x \in \mathbb{R}^V} \frac{(x_u - x_v)^2}{x^\top L_G x} \geq \frac{(x_u^0 - x_v^0)^2}{x^{0\top} L_G x^0},$$

where x^0 is 0 on the connected component of u in G_+, and 1 everywhere else (including v). Now

$$\frac{(x_u^0 - x_v^0)^2}{x^{0\top} L_G x^0} = \frac{1}{x^{0\top} L_{G_+} x^0 + x^{0\top} L_{G_-} x^0} = \frac{1}{x^{0\top} L_{G_-} x^0} = \frac{1}{\sum_{(a,b) \in E_-} z_-(a,b) \cdot (x_a^0 - x_b^0)^2} \geq \frac{1}{\zeta}.$$

This is a contradiction; therefore, u and v are indeed in the same connected component of G_+.

We can now apply Lemma 7.8:

$$\frac{1}{R_{G_+}(u,v)} \geq \frac{1}{R_G(u,v)} - \zeta \geq \frac{4}{5} \cdot \frac{1}{R_G(u,v)},$$

as claimed. $\qquad\square$

We are now ready to analyze Algorithm 6.

Theorem 7.10. *Let $H = (V, E, w)$ be an (α, β)-separated weighted hypergraph with $\beta \geq 5|E| \cdot \gamma/\eta$. Let $M = w(E)$ be the total hyperedge weight in H, and let $w_{\min}$ the minimum hyperedge weight. Then for $\gamma \geq 4$, Algorithm 6 terminates within $\alpha/\eta \cdot |E|^2 \cdot \mathrm{poly}\left(n \log(M/w_{\min})\right)$ rounds and returns an η-approximate 2γ-balanced weight assignment for H.*

Proof. We call executions of the while-loop rounds and executions of the for-loop phases. It will be useful to denote $w_{i,\min} := \min_{e \in E_i} w(e)$ and $w_{i,\max} := \max_{e \in E_i} w(e)$. The definition of separated hypergraphs (Definition 7.7) then ensures that $w_{i,\max} \leq \alpha \cdot w_{i,\min}$ and $w_{i,\min} \geq \beta \cdot w_{i+1,\max}$.

Algorithm Correctness. We first prove that upon termination, Algorithm 6 indeed returns an η-approximate 2γ-balanced weight assignment. Consider a hyperedge e and edges $f, g \in F_e$, with $z(g) \geq \eta \cdot w(e)$. We will show that $R_{G^*}(f) \leq 2\gamma \cdot R_{G^*}(g)$, where G^* is the final weight assignment returned by the algorithm at Line 10. This is exactly what is required by the definition of approximate balancedness, i.e. Definition 5.9. Suppose $e \in E_i$. Note that by the condition of the while-loop, at the termination of the ith phase, e, f, and g satisfied even the stronger condition of η-approximate γ-balancedness. Let us denote by G' the weight assignment graph at the end of phase i.

Let us partition the edges of G^* and G' based on the weight of the corresponding hyperedge. Denote $E_{\leq i} := \bigcup_{j=1}^{i} E_j$ and $E_{>i} := \bigcup_{j=i+1}^{\ell} E_j$. Similarly, let $F_{\leq i} := \bigcup_{e \in E_{\leq i}} F_e$ and $F_{>i} := \bigcup_{e \in E_{>i}} F_e$. Finally, let $G_{\leq i}^*$ and $G_{>i}^*$ be G^* restricted to $F_{\leq i}$ and $F_{>i}$ respectively, and define $G_{\leq i}'$ and $G_{>i}'$ similarly. Note that $G_{\leq i}^* = G_{\leq i}'$, since the algorithm never alters the weight assignments of $E_{\leq i}$ after phase i.

From here, we will use two applications of Corollary 7.9. First, set $G_+ := G_{\leq i}'$, $G_- := G_{>i}'$ (which makes $G = G'$), and $(u,v) := f$. Then ζ is the total weight of all edges in $G_{>i}'$, which is the total weight of all hyperedges in $E_{>i}$, which is at most $w_{i+1,\max} \cdot |E|$. We verify the condition on $R_{G'}(f)$ from Corollary 7.9: $R_{G'}(f) \leq \gamma \cdot R_{G'}(g)$, by the approximate γ-balancedness condition on e in G'. Since $z(g) \geq \eta \cdot w(e)$ (in both G^* and G'), we have that $z(g) \geq \eta \cdot w_{i,\min}$, and hence by Fact 2.8

$R_{G'}(g) \le 1/(\eta \cdot w_{i,\min})$. Finally, putting these together, as well as using the (α, β)-separated quality of H, we get

$$R_{G'}(f) \le \frac{\gamma}{\eta \cdot w_{i,\min}} \le \frac{\gamma}{\eta\beta \cdot w_{i+1,\max}} \le \frac{1}{5\zeta},$$

by assumption on β. We can indeed apply Corollary 7.9, which gives us

$$R_{G'}(f) \ge \frac{4}{5} \cdot R_{G_{\le i}}(f). \tag{18}$$

We apply Corollary 7.9 again, this time setting $G_+ := G^*_{\le i}$, $G_- := G^*_{>i}$ (which makes $G = G^*$), and $(u, v) := g$. Then $\zeta \le w_{i+1,\max} \cdot |E|$ by an identical argument to the previous setting. We verify the condition on $R_{G^*}(g)$ from Corollary 7.9: $R_{G^*}(g) \le 1/(\eta \cdot w_{i,\min})$ by Fact 2.8, since $z(g) \ge \eta \cdot w(e) \ge \eta \cdot w_{i,\min}$ by assumption (in both G^* and G'). Therefore, by an identical derivation to the previous case $R_{G^*}(g) \le 1/(5\zeta)$. We can indeed apply Corollary 7.9, which gives us

$$R_{G^*}(g) \ge \frac{4}{5} \cdot R_{G_{\le i}}(g). \tag{19}$$

Putting together Equations (18) and (19), along with the fact that f and g satisfied the condition of approximate γ-balancedness at the time of G' (that is at the end of phase i), we get

$$R_{G^*}(f) \le R_{G_{\le i}}(f) \le \frac{5}{4} \cdot R_{G'}(f) \le \frac{5\gamma}{4} \cdot R_{G'}(g) \le \frac{5\gamma}{4} \cdot R_{G_{\le i}}(g) \le \frac{25\gamma}{16} \cdot R_{G^*}(g) \le 2\gamma \cdot R_{G^*}(g),$$

which concludes the proof that G^* is η-approximately 2γ-balanced.

Running Time. Next, we prove that Algorithm 6 terminates within $\alpha/\eta \cdot |E|^2 \cdot \text{poly}(n \log(M/w_{\min}))$ rounds. Similarly to the proof of Theorem 5.10, we can bound the total growth of the ST-potential over the course of the algorithm by defining G_0 and G_∞. Specifically, if G_0 is the starting graph of Algorithm 6 and G_∞ is the complete graph of uniform edge weight M, then as before,

$$\Psi(G_\infty) - \Psi(G_0) \le \text{poly}(n \log(M/w_{\min})).$$

We will now prove that the total number of rounds in the ith phase is at most $|E_i| \cdot \alpha/\eta \cdot |E| \cdot \text{poly}(n \log(M/w_{\min}))$. Due to Lemma 5.7, the contribution of each update to the potential $\Psi(G)$ is at least

$$\log\left(1 + \lambda\gamma \cdot R_G(g) - \lambda \cdot R_G(g) - \lambda^2\gamma \cdot R_G(g)^2\right).$$

Similarly to the proof of Theorem 5.10, this either means that $\lambda = 1/(2R_G(g))$ and therefore the increase to Ψ is at least $\log(5/4)$, or that $\lambda = z(g) \le 1/(2R_G(g))$, in which case the increase to Ψ is at least $\log(1 + z(g) \cdot R_G(g))$. In the latter case, we further distinguish based on the value of $R_G(g)$.

1. $\lambda = 1/(2R_G(g))$. In this case the ST-potential increases by at least $\log(5/4)$, and so there can be at most $\text{poly}(n \log(M/w_{\min}))$ such rounds.

2. $\lambda = z(g)$ and $R_G(g) > 1/(5w_{i,\max} \cdot |E|)$. In this case the ST-potential increases by at least $\log(1 + z(g) \cdot R_G(g))$. Since g was selected in this round, $z(g) \ge \eta \cdot w(e) \ge \eta \cdot w_{i,\min}$, and therefore the increase in $\Psi(G)$ is at least

$$\log(1 + z(g) \cdot R_G(g)) \ge \log\left(1 + \frac{\eta \cdot w_{i,\min}}{5w_{i,\max} \cdot |E|}\right) \ge \log\left(1 + \frac{\eta}{5\alpha \cdot |E|}\right) \ge \frac{\eta}{10\alpha \cdot |E|}.$$

Thus, there can be at most $\alpha/\eta \cdot |E| \cdot \text{poly}(n \log(M/w_{\min}))$ such rounds.

3. $\lambda = z(g)$ and $R_G(g) \le 1/(5w_{i,\max} \cdot |E|)$. This is the most complicated case to analyze and we will be focusing on this for the rest of the proof.

Suppose we are in the ith phase of Algorithm 6 and we have $e \in E_i$ and $f, g \in F_e$ such that $R_G(f) > \gamma \cdot R_G(g) \ge 4 \cdot R_G(g)$. Further suppose that $R_G(g) \le 1/(5w_{i,\max} \cdot |E|)$. Similarly to before, we divide G into two graphs: $G_{<i}$ consisting of edges belonging to hyperedges from E_j for $j < i$, and $G_{\ge i}$ consisting of edges belonging to hyperedges from E_j for $j \ge i$. (Note that earlier in this proof we used the slightly different split into $G_{\le i}$ and $G_{>i}$.) We can again use Corollary 7.9 to relate R_G to $R_{G_{<i}}$.

We apply Corollary 7.9, setting $G_+ := G_{<i}$, $G_- := G_{\ge i}$ (which makes G from Corollary 7.9 the current graph G), and $(u, v) := g$. Then ζ is the total weight of all edges in $G_{\ge i}$, which is the total weight of all hyperedges in $E_{\ge i}$, which is at most $w_{i,\max} \cdot |E|$. Our above assumption $R_G(g) \le 1/(5w_{i,\max} \cdot |E|)$ exactly guarantees that $R_G(g) \le 1/(5\zeta)$ is satisfied and we can therefore apply Corollary 7.9:

$$R_{G_{<i}}(g) \le \frac{5}{4} \cdot R_G(g) \le \frac{5}{4\gamma} R_G(f) \le \frac{5}{16} R_G(f) \le \frac{5}{16} R_{G_{<i}}(g) < R_{G_{<i}}(f).$$

In words, such weight transfers are always from edges of lower $G_{<i}$-resistance to those of strictly higher $G_{<i}$-resistance—a metric that never changes, since $G_{<i}$ is unchanged during the ith stage of the algorithm. Therefore, there cannot be $n^4 \cdot |E_i|$ consecutive steps of type 3 in phase i. We prove this formally in the following claim.

Claim 7.11. *There cannot be $n^4 \cdot |E_i|$ consecutive rounds of type 3 in phase i.*

Proof. Suppose for contradiction that there is a sequence of $n^4 \cdot |E_i|$ rounds of type 3. Then there must be a hyperedge $e \in E_i$ for which at least n^4 of these updates take place in F_e. We know that updates can only shift weight from g to f where $R_{G_{<i}}(f) > R_{G_{<i}}(g)$. Therefore, let us order the edges of F_e by $R_{G_{<i}}$, that is, let $\rho : F_e \to \mathbb{N}$ be such that the jth largest edge in terms of $R_{G_{<i}}$, say f, has $\rho(f) = j$ (breaking ties arbitrarily). Then we can define a local potential function for e:

$$\psi_e(G) = \sum_{f \in F_e} \rho(f) \cdot \mathbb{1}(z(f) > 0).$$

Note that ψ_e starts out as at most $1 + 2 + \cdots + |F_e| \le n^4$, and after every update of type 3 it decreases by at least one. Since we have no updates of any type other than 3 (by assumption), and updates to other hyperedges do not affect ψ_e, we arrive at the contradiction that ψ_e must become negative. $\qquad\square$

As a result of Claim 7.11, every consecutive sequence of $n^4 \cdot |E_i|$ rounds must contain an update of type 1 or 2. Since we showed that there are at most $\alpha/\eta \cdot |E| \cdot \text{poly}\,(n \log(M/w_{\min}))$ such updates, there can be at most $|E_i| \cdot \alpha/\eta \cdot |E| \cdot \text{poly}\,(n \log(M/w_{\min}))$ rounds in phase i. Summing this over all phases, we get that there can be at most $\alpha/\eta \cdot |E|^2 \cdot \text{poly}\,(n \log(M/w_{\min}))$ rounds overall, as claimed. $\qquad\square$

7.3 Proof of the Second Part of Theorem 1

We are now ready to prove the second part of Theorem 1. Specifically, we prove that FASTSPAR-SIFICATION (Algorithm 7 below) provides the result of Theorem 1:

Algorithm 7 Algorithm constructing nearly linear-sized spectral sparsifier, in nearly linear time.

1: **procedure** FASTSPARSIFICATION($H = (V, E, w)$)
2:      $H' = (V, E', w') \leftarrow$ POLYNOMIALSIZESPARSIFICATION($H, \epsilon/3$)   ▷ Algorithm 5
3:      **for** i from 1 to n **do**
4:          $E'_i \leftarrow \{e \in E'_i \mid w'(e) \in [n^{10(i-1)}, n^{10i})\}$
5:      $H'_1 \leftarrow (V, \bigcup_{j \in \{1,...,n\}, j \text{ odd}} E_i, w')$
6:      $H'_2 \leftarrow (V, \bigcup_{j \in \{1,...,n\}, j \text{ even}} E_i, w')$
7:      $G_1 \leftarrow$ SEPARATEDAPPROXBALANCING($H'_1, (E_j)_{j \text{ odd}}, n^{10}, n^{10}, 4, 1/n^2$)   ▷ Algorithm 6
8:      $G_2 \leftarrow$ SEPARATEDAPPROXBALANCING($H'_2, (E_j)_{j \text{ even}}, n^{10}, n^{10}, 4, 1/n^2$)
9:      $\lambda \leftarrow \Theta(\epsilon^{-4} \log^3 n)$
10:      $\widetilde{H}_1 \leftarrow$ SPARSIFICATION($H'_1, G_1, \epsilon/3, \lambda$)   ▷ Algorithm 3
11:      $\widetilde{H}_2 \leftarrow$ SPARSIFICATION($H'_2, G_2, \epsilon/3, \lambda$)
12:      **return** $\widetilde{H} \leftarrow \widetilde{H}_1 \cup \widetilde{H}_2$

In the final line $\widetilde{H} \leftarrow \widetilde{H}_1 \cup \widetilde{H}_2$ means that we take the union of hyperedges and weight functions from $\widetilde{H}_1$ and $\widetilde{H}_2$, since both are on the same vertex set V.

Proof of the second part of Theorem 1: Indeed, Algorithm 7 does exactly that.

First note a few observations about the steps of Algorithm 7: Notice that POLYNOMIALSIZES-PARSIFICATION produces hyperedge weights only in the range $[1, n^{10n}]$, so E' is partitioned into E'_i for i from 1 to n. Next, H'_1 and H'_2 are indeed (n^{10}, n^{10})-separated hypergraphs by the definitions in Lines 4, 5, and 6. Furthermore, note that $\beta \geq 5|E| \cdot \gamma/\eta$ holds (as required by Theorem 7.10), since $\beta = n^{10}$, $\eta = 1/n^2$, and $|E| = o(n^8)$ by the size guarantee of Corollary 7.6. Finally, G_1 and G_2 are $1/n^2$-approximate 8-balanced weight assignments of H'_1 and H'_2 respectively, by Theorem 7.10.

Therefore, by Lemma 6.2, $\widetilde{H}_1$ and $\widetilde{H}_2$ are $\epsilon/3$-spectral sparsifiers of H'_1 and H'_2 respectively. By the additivity of the hypergraph quadratic form $\widetilde{H}$ is an $\epsilon/3$-spectral sparsifier of H'. Since H' is itself an $\epsilon/3$-spectral sparsifier of H, this means that $\widetilde{H}$ is an ϵ-spectral sparsifier of H, as claimed.

Moreover, by Lemma 6.1, $\widetilde{H}_1$ and $\widetilde{H}_2$ are both of size at most $O(n \log^3(n)/\epsilon^4)$ (since we set λ to be $\Theta(\log^3(n)/\epsilon^4)$), and therefore so is $\widetilde{H}$.

Finally, Algorithm 7 runs in time $\widetilde{O}(mr + \text{poly}(n))$. Indeed POLYNOMIALSIZESPARSIFICATION runs in time $\widetilde{O}(mr) + \text{poly}(n)$, as shown in Lemma 7.6; SEPARATEDAPPROXBALANCING runs in time $\text{poly}(n)$ by Theorem 7.10, since $\alpha = \text{poly}(n)$ and $M = \exp(O(n))$; SPARSIFICATION runs in time $\text{poly}(n)$, since the input hypergraph is of $\text{poly}(n)$ size. $\square$

References

[ACK+16]   Alexandr Andoni, Jiecao Chen, Robert Krauthgamer, Bo Qin, David P. Woodruff, and Qin Zhang. On sketching quadratic forms. In *Proceedings of the 2016 Conference on Innovations in Theoretical Computer Science (ITCS)*, pages 311–319, 2016. doi:10.1145/2840728.2840753.

[AK17]   N. Alon and B. Klartag. Optimal compression of approximate inner products and dimension reduction. In *Proceedings of the IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 639–650, 2017. doi:10.1109/FOCS.2017.65.

[AS08]   Noga Alon and Joel H. Spencer. *The Probabilistic Method, Third Edition.* Wiley-Interscience series in discrete mathematics and optimization. Wiley, 2008.

[BK15]   András A. Benczúr and David R. Karger. Randomized approximation schemes for cuts
            and flows in capacitated graphs. *SIAM Journal on Computing*, 44(2):290–319, 2015.
            `doi:10.1137/070705970`.

[BSS12]   Joshua D. Batson, Daniel A. Spielman, and Nikhil Srivastava. Twice-Ramanujan spar-
            sifiers. *SIAM Journal on Computing*, 41(6):1704–1721, 2012. `doi:10.1137/090772873`.

[BST19]   Nikhil Bansal, Ola Svensson, and Luca Trevisan. New notions and constructions of
            sparsification for graphs and hypergraphs. In *Proceedings of the IEEE 60th Annual
            Symposium on Foundations of Computer Science (FOCS)*, pages 910–928, 2019. `doi:
            10.1109/focs.2019.00059`.

[CKN20]   Yu Chen, Sanjeev Khanna, and Ansh Nagda. Near-linear size hypergraph cut sparsi-
            fiers. In *Proceedings of the 61st IEEE Annual Symposium on Foundations of Computer
            Science, FOCS 2020*, pages 61–72, 2020. `doi:10.1109/FOCS46700.2020.00015`.

[CKST19]   Charles Carlson, Alexandra Kolla, Nikhil Srivastava, and Luca Trevisan. Optimal lower
            bounds for sketching graph cuts. In *Proceedings of the 13th Annual ACM-SIAM Sym-
            posium on Discrete Algorithms (SODA)*, pages 2565–2569, 2019.

[CLTZ18]   T.-H. Hubert Chan, Anand Louis, Zhihao Gavin Tang, and Chenzi Zhang. Spectral
            properties of hypergraph Laplacian and approximation algorithms. *Journal of the ACM*,
            65(3):1–48, 2018. `doi:10.1145/3178123`.

[FSY18]   Kaito Fujii, Tasuku Soma, and Yuichi Yoshida. Polynomial-time algorithms for sub-
            modular Laplacian systems, 2018. `arXiv:1803.10923`.

[HSJR13]   Matthias Hein, Simon Setzer, Leonardo Jost, and Syama Sundar Rangapuram. The
            total variation on hypergraphs - learning on hypergraphs revisited. In *Advances in
            Neural Information Processing Systems 26 (NIPS)*, pages 2427–2435, 2013.

[IKT21]   Masahiro Ikeda, Yu Kitabeppu, and Yuuki Takai. Coarse Ricci curvature of hypergraphs
            and its generalization, 2021. `arXiv:2102.00698`.

[IMTY19]   Masahiro Ikeda, Atsushi Miyauchi, Yuuki Takai, and Yuichi Yoshida. Finding Cheeger
            cuts in hypergraphs via heat equation, 2019. `arXiv:1809.04396`.

[KK15]   Dmitry Kogan and Robert Krauthgamer. Sketching cuts in graphs and hypergraphs.
            In *Proceedings of the 2015 Conference on Innovations in Theoretical Computer Science
            (ITCS)*, pages 367–376, 2015. `doi:10.1145/2688073.2688093`.

[KKTY21]   Michael Kapralov, Robert Krauthgamer, Jakab Tardos, and Yuichi Yoshida. Towards
            tight bounds for spectral sparsification of hypergraphs. In *Proceedings of the 53rd Annual
            ACM Symposium on Theory of Computing (STOC)*, 2021. to appear.

[Kom67]   Yukio Komura. Nonlinear semi-groups in Hilbert space. *Journal of the Mathematical
            Society of Japan*, 19(4):493–507, 1967.

[LM18]   Pan Li and Olgica Milenkovic. Submodular hypergraphs: p-Laplacians, Cheeger in-
            equalities and spectral clustering. In *Proceedings of the 35th International Conference
            on Machine Learning (ICML)*, volume 80, pages 3020–3029, 2018.

[Lou15]   Anand Louis. Hypergraph Markov operators, eigenvalues and approximation algorithms. In *Proceedings of the 47th Annual ACM on Symposium on Theory of Computing (STOC)*, pages 713–722, 2015. doi:10.1145/2746539.2746555.

[Lov93]   László Lovász. Random walks on graphs: A survey. In D. Miklós, Vera T. Sós, and T. Szönyi, editors, *Combinatorics, Paul ErdHos is eighty. Vol. 2*, pages 353–397. János Bolyai Mathematical Society, 1993. URL: http://bolyai.math.elte.hu/~lovasz/erdos.pdf.

[LS15]   Yin Tat Lee and He Sun. Constructing linear-sized spectral sparsification in almost-linear time. In Venkatesan Guruswami, editor, *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, pages 250–269. IEEE Computer Society, 2015. doi:10.1109/FOCS.2015.24.

[LS17]   Yin Tat Lee and He Sun. An sdp-based algorithm for linear-sized spectral sparsification. In Hamed Hatami, Pierre McKenzie, and Valerie King, editors, *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 678–687. ACM, 2017. doi:10.1145/3055399.3055477.

[Mey73]   C. Meyer. Generalized inversion of modified matrices. *SIAM Journal on Applied Mathematics*, 24(3):315–323, 1973.

[Miy92]   Isao Miyadera. *Nonlinear Semigroups*, volume 109. American Mathematical Soc., 1992.

[NR13]   Ilan Newman and Yuri Rabinovich. On multiplicative λ-approximations and some geometric applications. *SIAM Journal on Computing*, 42(3):855–883, 2013. doi:10.1137/100801809.

[RV07]   Mark Rudelson and Roman Vershynin. Sampling from large matrices: An approach through geometric functional analysis. *Journal of the ACM*, 54(4):21, 2007. doi:10.1145/1255443.1255449.

[SS11]   Daniel A. Spielman and Nikhil Srivastava. Graph sparsification by effective resistances. *SIAM Journal on Computing*, 40(6):1913–1926, 2011. doi:10.1137/080734029.

[ST11]   Daniel A. Spielman and Shang-Hua Teng. Spectral sparsification of graphs. *SIAM Journal on Computing*, 40(4):981–1025, 2011. doi:10.1137/08074489x.

[SY19]   Tasuku Soma and Yuichi Yoshida. Spectral sparsification of hypergraphs. In *Proceedings of the 30th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2570–2581. jan 2019. doi:10.1137/1.9781611975482.159.

[Ten16]   Shang-Hua Teng. Scalable algorithms for data and network analysis. *Foundations and Trends® in Theoretical Computer Science*, 12(1-2):1–274, 2016. doi:10.1561/0400000051.

[TMIY20] Yuuki Takai, Atsushi Miyauchi, Masahiro Ikeda, and Yuichi Yoshida. Hypergraph clustering based on PageRank. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*, pages 1970–1978, 2020. doi:10.1145/3394486.3403248.

[Tro11]   Joel A. Tropp. User-friendly tail bounds for sums of random matrices. *Foundations of Computational Mathematics*, 12(4):389–434, 2011. `doi:10.1007/s10208-011-9099-z`.

[Vis13]   Nisheeth K. Vishnoi. Lx = b. *Foundations and Trends® in Theoretical Computer Science*, 8(1–2):1–141, 2013. `doi:10.1561/0400000054`.

[YNN⁺20]   Naganand Yadati, Vikram Nitin, Madhav Nimishakavi, Prateek Yadav, Anand Louis, and Partha Talukdar. NHP: Neural hypergraph link prediction. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management (CIKM)*, pages 1705–1714, 2020. `doi:10.1145/3340531.3411870`.

[YNY⁺19]   Naganand Yadati, Madhav Nimishakavi, Prateek Yadav, Vikram Nitin, Anand Louis, and Partha P. Talukdar. HyperGCN: A new method for training graph convolutional networks on hypergraphs. In *Advances in Neural Information Processing Systems 32*, pages 1509–1520, 2019.

[Yos16]   Yuichi Yoshida. Nonlinear Laplacian for digraphs and its applications to network analysis. In *Proceedings of the 9th ACM International Conference on Web Search and Data Mining (WSDM)*, pages 483–492, 2016. `doi:10.1145/2835776.2835785`.

[Yos19]   Yuichi Yoshida. Cheeger inequalities for submodular transformations. In *Proceedings of the 30th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2582–2601, 2019. `doi:10.1137/1.9781611975482.160`.

[ZHTC20]   Chenzi Zhang, Shuguang Hu, Zhihao Gavin Tang, and T-H. Hubert Chan. Re-revisiting learning on hypergraphs: Confidence interval, subgradient method, and extension to multiclass. *IEEE Transactions on Knowledge and Data Engineering*, 32(3):506–518, 2020. `doi:10.1109/tkde.2018.2880448`.

[ZLO15]   Zeyuan Allen Zhu, Zhenyu Liao, and Lorenzo Orecchia. Spectral sparsification and regret minimization beyond matrix multiplicative updates. In Rocco A. Servedio and Ronitt Rubinfeld, editors, *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 237–245. ACM, 2015. `doi:10.1145/2746539.2746610`.