# SPECTRAL SPARSIFICATION OF SIMPLICIAL COMPLEXES FOR CLUSTERING AND LABEL PROPAGATION[*]

*Braxton Osting,*[†] *Sourabh Palande,*[‡] *and Bei Wang*[§]

Abstract. As a generalization of the use of graphs to describe pairwise interactions, simplicial complexes can be used to model higher order interactions between three or more objects in complex systems. There has been a recent surge in activity to develop data analysis methods applicable to simplicial complexes, including techniques based on computational topology, higher order random processes, generalized Cheeger inequalities, isoperimetric inequalities, and spectral methods. In particular, spectral learning methods (e.g., label propagation and clustering) that directly operate on simplicial complexes represent a new direction for analyzing such complex datasets.

To apply spectral learning methods to massive datasets modeled as simplicial complexes, we develop a method for sparsifying simplicial complexes that preserves the spectrum of the associated Laplacian matrices. We show that the theory of Spielman and Srivastava for the sparsification of graphs extends to simplicial complexes via the up Laplacian. In particular, we introduce a generalized effective resistance for simplices, provide an algorithm for sparsifying simplicial complexes at a fixed dimension, and give a specific version of the generalized Cheeger inequality for weighted simplicial complexes. Finally, we introduce higher order generalizations of spectral clustering and label propagation for simplicial complexes and demonstrate via experiments the utility of the proposed spectral sparsification method for these applications.

## 1 Introduction

Understanding massive systems with complex interactions and multiscale dynamics is important in a variety of social, biological, and technological settings. A common approach to understanding such a system is to represent it as a graph where vertices represent objects, and (weighted) edges represent *pairwise interactions* between the objects. A large arsenal of methods has been developed to analyze properties of graphs, which can then be combined with domain-specific knowledge to infer properties of the system being studied. These tools include graph partitioning and clustering [54, 70, 71], random processes on graphs [32], graph distances, various measures of graph connectivity [53], combinatorial graph invariants [23],

[*] This work was partially supported by NSF DMS-1461138, NSF DMS-1752202, NSF IIS-1513616, and the University of Utah Seed Grant 10041533. We would like to thank Todd H. Reeb for contributing to early discussions.

[†] *University of Utah*, osting@math.utah.edu

[‡] *University of Utah*, sourabh@sci.utah.edu

[§] *University of Utah*, beiwang@sci.utah.edu

and spectral graph theory [15]. In particular, spectral methods for graph-based learning have had great success due to their efficiency and good theoretical guarantees for applications ranging from image segmentation [47] to community detection [2]. For example, the spectral clustering method (see, e.g., [1, 65]) is a graph-based learning method used for the unsupervised clustering task, and label propagation [68, 76] is a graph-based learning method for semisupervised regression.

**Simplicial complexes and data analysis.** Although graphs have been used with great success to describe pairwise interactions between objects in datasets, they fail to capture *higher order interactions* that occur between three or more objects. Higher order interactions in complex datasets can be modeled using *simplicial complexes* [34, 49]. The recent surge in activity to develop machine learning methods for data represented by simplicial complexes has included methods based on computational topology [10, 27, 30, 34], higher order random processes [7, 33], generalized Cheeger inequalities [35, 67], isoperimetric inequalities [57], high-dimensional expanders [24, 46, 56], and spectral methods [37]. In particular, topological data analysis methods using simplicial complexes as the underlying combinatorial structures have been successfully employed for diverse applications [20, 38, 43, 52, 55, 58, 59, 72].

Learning (indirectly or directly) based on simplicial complexes represents a new direction that has recently emerged from the confluence of computational topology and machine learning. This work is ongoing; whereas topological features derived from simplicial complexes, used as input to machine learning algorithms, have been shown to increase the predictive power compared to graph-theoretic features [6, 74], interest in developing learning algorithms that directly operate on simplicial complexes is growing. For example, researchers have begun to develop mathematical intuition behind higher dimensional notions of spectral clustering and label propagation [48, 67, 71].

**Sparsification of graphs and simplicial complexes.** For unstructured graphs representing massive datasets, the computational costs associated with naïve implementations of many graph based algorithms are prohibitive. In this scenario, it is useful to approximate the original graph with one having fewer edges or vertices while preserving certain properties of interest, known as *graph sparsification*. A variety of graph sparsification methods have been developed that allow for both efficient storage and computation [5, 63, 64]; see [4] for a survey. In particular, Spielman and Srivastava developed a method for graph sparsification using effective resistances of edges that approximately preserves the spectrum of the graph Laplacian [63]; their work is most relevant to the results presented in this paper. It is well known from spectral graph theory that the spectrum of the graph Laplacian bounds a variety of properties of interest, including the size of cuts (i.e., bottlenecks), clusters (i.e., communities), distances, various random processes (i.e., PageRank), and combinatorial properties (e.g., coloring, spanning trees, etc.). It follows that this method [4] can be used to produce a sparsified graph that contains a great deal of information about the original graph and, hence, in the graph-based machine learning setting, about the underlying dataset.

Analogously, computational methods that operate on simplicial complexes are severely limited by the computational costs associated with massive datasets. Although geometric complexes (embedded in Euclidean space) tend to be naturally sparse, abstract simplicial complexes coming from data analysis can be dense and do not have natural embeddings in

Euclidean space. For example, a dense simplicial complex is obtained when representing functional brain networks using simplicial complexes (e.g., [11, 44, 45]). Here, a brain network is mapped to a point cloud in a metric space, where network nodes map to points, pairwise associations between nodes map to distances between pairs of points, and higher order information is mapped to higher dimensional simplices [3, 74]. Another motivation for studying sparsification of simplicial complexes is the fact that high-order tensors (multidimensional arrays) can be represented by simplicial complexes and vice versa. Just as spectral graph sparsifiers are useful in matrix decompositions and linear system solvers, one can expect simplicial complex sparsifiers to be useful in tensor decompositions and multilinear system solvers.

Several approaches have recently been proposed to sparsify simplicial complexes. One class of methods, referred to as *homological sparsification*, involves constructing a sparse simplicial complex that approximates persistence homology [8, 9, 12, 14, 21, 22, 40, 61, 69]. Persistence homology [28] turns the algebraic concept of homology into a multiscale notion. It typically operates on a sequence of simplicial complexes (referred to as a *filtration*), constructs a series of homology groups, and measures their relevant scales in the filtration. Common simplicial filtrations arise from Čech or Vietoris-Rips complexes, and most of the homological sparsification techniques produce sparsified complexes that give guaranteed approximations to the persistent homology of the unsparsified filtration.

The sparsification processes involve either the removal or subsampling of vertices or edge contraction from the sparse filtration. It is also possible to sparsify simplicial complexes using another class of methods called sketching, particularly those applied to tensors. Tensor decomposition methods have found many applications in machine learning [41], including recent advancements in tensor sparsification [73, 36, 51, 62] using sampling methods from randomized linear algebra.

Since many learning methods based on simplicial complexes rely — either explicitly or implicitly — on the spectral theory for higher order Laplacians, developing methods for sparsifying simplicial complexes that approximately preserves the spectrum of higher order Laplacians is desirable.

**Contributions.** In this paper, motivated by learning based on simplicial complexes, we develop computational methods for the spectral sparsification of simplicial complexes. In particular:

- We introduce a *generalized effective resistance* of simplices by extending the notion of *effective resistance* of edges (e.g., [13, 25, 29]); see Section 3.1.
- We extend the methods and analysis of Spielman and Srivastava [63] for sparsifying graphs to the context of simiplicial complexes at a fixed dimension. We prove that the spectrum of the *up Laplacian* is approximately preserved under sparsification in the sense that the spectrum of the up Laplacian for the sparsified simplicial complex is controlled by the spectrum of the up Laplacian for the original simplicial complex; see Theorem 3.1.
- We generalize the Cheeger constant of Gundert and Szedlák for *unweighted* simplicial complexes [35] to *weighted* simplicial complexes and show that the Cheeger constant of the sparsified simplicial complex is bounded below by a multiplicative factor of the first

nontrivial eigenvalue of the up Laplacian for the original complex; see Corollary 4.1.

- Our theoretical results are supported by substantial numerical experiments. By extending spectral learning algorithms such as spectral clustering and label propagation to simplicial complexes, we demonstrate that preserving the structure of the up Laplacian via sparsification also preserves the results of these algorithms (Section 5). These applications exemplify the utility of our spectral sparsification methods.

We proceed as follows: In Section 2 we introduce the notation and give a brief description of relevant algebraic concepts such as effective resistance and spectral sparsification of graphs. The theory and algorithm for sparsifying simplicial complexes are presented in Section 3. We state the implications of the algorithm for a generalized Cheeger cut for the simplicial complex in Section 4. We showcase experimental results validating our algorithms in Section 5 and conclude with a discussion and some open questions in Section 6.

## 2 Background

**Simplicial complexes.** A *simplicial complex* $K$ is a finite collection of simplices such that every face of a simplex of $K$ is in $K$, and the intersection of any two simplices of $K$ is a face of each of them [49]. The 0-, 1-, and 2-simplices correspond to vertices, edges, and triangles. An *oriented simplex* is a simplex with a chosen ordering of its vertices. Consider an oriented $(k+1)$-simplex $\tau = [v_0, \ldots, v_{k+1}]$ of $K$ where $v_0 < \cdots < v_{k+1}$ is the vertex ordering. $\sigma = \tau \backslash \{v_j\}$ is the $k$-simplex obtained from $\tau$ by omitting vertex $v_j$. The *oriented incidence number* $[\tau \colon \sigma]$ of a $k$-simplex $\sigma$ of $K$ is defined as $(-1)^j$ if $\sigma = \tau \backslash \{v_j\}$ for some $j = 0, \ldots, k+1$ and 0 if $\sigma \nsubseteq \tau$. For the remainder of this paper, we will assume $K$ is an oriented simplicial complex on a vertex set $V = \{v_1, v_2, \ldots, v_n\}$. $S_k(K)$ denotes the collection of all oriented $k$-simplices of $K$ and $n_k = |S_k(K)|$. The *p-skeleton* of $K$ is denoted as $K^{(p)} := \bigcup_{0 \le k \le p} S_k(K)$. Let $\dim(K)$ denote the dimension of $K$. For a review of simplicial complexes, see [31, 34, 49].

**Laplace operators on simplicial complexes.** The $k$-th *chain group* $C_k(K) = C_k(K, \mathbb{R})$ of a complex $K$ with coefficient $\mathbb{R}$ is a vector space over the field $\mathbb{R}$ with basis $S_k(K)$. The $k$-th *cochain group* $C^k(K) = C^k(K, \mathbb{R})$ is the dual of the chain group, defined by $C^k(K) := \mathrm{Hom}(C_k(K), \mathbb{R})$, where $\mathrm{Hom}(C_k(K), \mathbb{R})$ denotes all homomorphisms of $C_k(K)$ into $\mathbb{R}$. The coboundary operator $\delta_k \colon C^k(K) \to C^{k+1}(K)$ is defined as $(\delta_k f)(\tau) = \sum_{\sigma \in S_k} [\tau \colon \sigma] f(\sigma)$. Let $Z^k = \mathrm{Ker}(\delta_k)$ and $B^k = \mathrm{Im}(\delta_{k-1})$ denote the groups of $k$-dimensional *cocycles* and $k$-dimensional *coboundaries*, respectively. The coboundary operator satisfies the property $\delta_k \delta_{k-1} = 0$, which implies that $B^k \subseteq Z^k$. The boundary operators, $\delta_k^*$, are the adjoints of the coboundary operators,

$$\begin{tikzcd}
\cdots & C^{k+1}(K) \arrow[r, shift right, "\delta_k^*"'] & C^k(K) \arrow[l, shift right, "\delta_k"'] \arrow[r, shift right, "\delta_{k-1}^*"'] & C^{k-1}(K) \arrow[l, shift right, "\delta_{k-1}"'] & \cdots
\end{tikzcd}$$

satisfying $(\delta_k a, b)_{C^{k+1}} = (a, \delta_k^* b)_{C^k}$ for every $a \in C^k(K)$ and $b \in C^{k+1}(K)$, where $(\cdot, \cdot)_{C^k}$ denotes the scalar product on the cochain group. We denote by $Z_k = \mathrm{Ker}(\delta_k^*)$ and $B_k = \mathrm{Im}(\delta_{k+1}^*)$, the groups of $k$-dimensional *cycles* and $k$-dimensional *boundaries*, respectively.

Following [37], we define three combinatorial Laplace operators that operate on $C^k(K)$ (for the $k$-th dimension), namely, the *up Laplacian*,

$$\mathcal{L}_k^{\text{up}}(K) = \delta_k^* \delta_k,$$

the *down Laplacian*, $\mathcal{L}_k^{\text{down}}(K) = \delta_{k-1}\delta_{k-1}^*$, and the *Laplacian*, $\mathcal{L}_k(K) = \mathcal{L}_k^{\text{up}}(K) + \mathcal{L}_k^{\text{down}}(K)$. All three operators are self-adjoint, non-negative, and compact, and they enjoy a collection of spectral properties, as detailed in [37]. We restrict our attention to the up Laplacians.

**Explicit expression for the up Laplacian.** To make the expression of the up Laplacian explicit, we need to choose a scalar product on the coboundary vector spaces that can be viewed in terms of weight functions [37]. In particular, the weight function $w$ is evaluated on the set of all simplices of $K$, $w\colon \bigcup_{k=0}^{\dim(K)} S_k(K) \to \mathbb{R}^+$, where the weight of a simplex $f$ is $w(f)$ (also denoted as $w_f$). Let $w_k\colon S_k(K) \to \mathbb{R}^+$. Then, $C^k(K)$ is the space of real-valued functions on $S_k(K)$, with inner product $(a,b)_{C^k} := \sum_{f \in S_k(K)} w_k(f)a(f)b(f)$, for every $a, b \in C^k(K)$.

Choosing the natural bases, we identify each coboundary operator $\delta_k$ with an incidence matrix $D_k$. The *incidence matrix* $D_k \in \mathbb{R}^{n_{k+1}} \times \mathbb{R}^{n_k}$ encodes which $k$-simplices are incident to which $(k+1)$ simplices in the complex, and is defined as

$$D_k(i,j) = \begin{cases} 0 & \text{if } \sigma_j^k \text{ is not on the boundary of } \sigma_i^{k+1} \\ 1 & \text{if } \sigma_j^k \text{ is coherent with the induced orientation of } \sigma_i^{k+1} \\ -1 & \text{if } \sigma_j^k \text{ is not coherent with the induced orientation of } \sigma_i^{k+1}. \end{cases}$$

Let $D_k^T$ be the transpose of $D_k$. Let $W_k$ be the diagonal matrix representing the scalar product on $C^k(K)$. The $k$-dimensional up Laplacian can then be expressed in the chosen bases as the matrix

$$\mathcal{L}_k^{\text{up}}(K) = W_k^{-1} D_k^T W_{k+1} D_k.$$

**Effective resistance.** We quickly review the notation in [63] regarding effective resistance. Let $G = (V, E, w)$ be a connected weighted undirected graph with $n$ vertices and $m$ edges, and edge weights $w_e \in \mathbb{R}^+$. $W$ is an $m \times m$ diagonal matrix with $W(e,e) = w_e$. Suppose the edges are oriented arbitrarily. The Laplacian $L \in \mathbb{R}^{n \times n}$ of $G$ can be written as

$$L = B^T W B,$$

where $B \in \mathbb{R}^{m \times n}$ is the signed edge-vertex incidence matrix, that is,

$$B(i,j) = \begin{cases} 0 & \text{if vertex } j \text{ is not on the boundary of edge } i \\ 1 & \text{if } j \text{ is } i\text{'s head} \\ -1 & \text{if } j \text{ is } i\text{'s tail.} \end{cases}$$

The effective resistance $R_e$ at an edge $e$ is the energy dissipation (potential difference) when a unit current is injected at one end and removed at the other end of $e$ [63]. Define the matrix $R := B(L)^+ B^T = B(B^T W B)^+ B^T$, where $L^+$ is the Moore-Penrose pseudoinverse of $L$. The diagonal entry $R(e,e)$ of $R$, is the effective resistance $R_e$ across $e$.

Using the previous notation for up Laplacians, we can write $L = D_0^T W_1 D_0 = W_0 \mathcal{L}_{K,0}$, by setting $B = D_0$ and $W = W_1$. Vertex weights are usually ignored in the graph sparsification literature, which is equivalent to setting the corresponding weight matrix $W_0$ to the identity. We can now express $R$ as $R_1 = D_0(L)^+ D_0^T = D_0(D_0^T W_1 D_0)^+ D_0^T$.

**Graph sparsification.** There are several different notions of approximation for graph sparsification, including the following based on spectral properties of the associated graph Laplacian. We say $H = (V, F, u)$ is a *sparse $\varepsilon$-approximation* of $G = (V, E, w)$ if $F \subset E$ and

$$(1 - \varepsilon) L_G \preceq L_H \preceq (1 + \varepsilon) L_G, \tag{1}$$

where $L_G$ and $L_H$ are the graph Laplacians of $G$ and $H$, respectively, and the inequalities are to be understood in the sense of the semidefinite matrix ordering. That is,

$$(1 - \varepsilon) x^T L_G x \le x^T L_H x \le (1 + \varepsilon) x^T L_G x \qquad \forall x \in \mathbb{R}^n.$$

## 3 Sparsification of simplicial complexes

We describe a sparsification algorithm for simplicial complexes and our main theoretical result (Theorem 3.1) in Section 3.1, whose proof is detailed in Section 3.2.

### 3.1 Sparsification algorithm

To prove the existence of a sparse $\varepsilon$-approximation of simplicial complex, we will follow the approach of [63] for the analogous problem for graphs.

**Generalized effective resistance for simplicial complexes.** We define the following symmetric positive semidefinite matrix:

$$\mathcal{L}_{K,k} = W_k \mathcal{L}_k^{\mathrm{up}}(K) = D_k^T W_{k+1} D_k.$$

To generalize the effective resistance for simplices beyond dimension one (i.e., edges), we consider the operator $R_k \colon C^k \to C^k$, defined by

$$R_k = D_{k-1}(\mathcal{L}_{K,k-1})^+ D_{k-1}^T = D_{k-1} \left( D_{k-1}^T W_k D_{k-1} \right)^+ D_{k-1}^T,$$

which is the projection onto the image of $D_{k-1}$. The *generalized effective resistance* of the $k$-dimensional simplex, $f$, is defined to be the diagonal entry, $R_k(f, f)$ (also denoted as $R_f$). For $k = 1$, the generalized effective resistance reduces to the effective resistance for graphs [29].

**Sparsification algorithm.** Algorithm 1 is a natural generalization of the **Sparsify** Algorithm given in [63]. The algorithm sparsifies a given simplicial complex $K$ at a fixed dimension $k$ (while ignoring all dimensions larger than $k$). The main idea is to include each $k$-simplex $f$ of $K$ in the sparsifier $J$ with a probability proportional to its generalized effective resistance. Specifially, for a fixed dimension $k$, the algorithm chooses a random $k$-simplex

```
Algorithm 1: $J = \textbf{Sparsify}(K, k, q)$
    Data: A weighted, oriented simplicial complex $K$, a dimension $k$ (where
          $1 \leq k \leq \dim(K)$), and an integer $q$.
    Result: A weighted, oriented simplicial complex $J$ that is sparsified at
            dimension $k$, with equivalent $(k-1)$-skeleton to $K$ and $\dim(J) = k$.
    $J := K^{(k-1)}$
    Sample $q$ $k$-dimensional simplices independently with replacement according to
      the probability
          $p_f = \frac{w_k(f)R_k(f,f)}{\sum_f w_k(f)R_k(f,f)}$,
      and add sampled simplices to $J$ with weight $w_k(f)/qp_f$. If a simplex is chosen
      more than once, the weights are summed.
```

$f$ of $K$ with probability $p_f$ (proportional to $w_f R_f$), and adds $f$ to $J$ with weight $w_f/qp_f$. $q$ samples are taken independently with replacement, summing the weights if a simplex is chosen more than once. The following theorem (Theorem 3.1) shows that if $q$ is sufficiently large, the $(k-1)$-dimensional up Laplacians of $K$ and $J$ are close:

**Theorem 3.1.** *Let $K$ be a weighted, oriented simpicial complex, and $J = \textbf{Sparsify}(K, k, q)$ for some fixed $k$ (where $1 \leq k \leq \dim(K)$). Suppose $K$ and $J$ have $(k-1)$-th up Laplacians $\mathcal{L}_K := \mathcal{L}^{up}_{k-1}(K)$ and $\mathcal{L}_J := \mathcal{L}^{up}_{k-1}(J)$, respectively. Let $n_{k-1}$ denote the number of $(k-1)$-simplices in $K$. Fix $\varepsilon > 0$ (where $1/\sqrt{n_{k-1}} < \varepsilon \leq 1$), and let $q = 9C^2 n_{k-1} \log n_{k-1}/\varepsilon^2$, where $C$ is an absolute constant. If $n_{k-1}$ is sufficently large, then with probability at least $1/2$,*

$$(1-\varepsilon)\mathcal{L}_K \preceq \mathcal{L}_J \preceq (1+\varepsilon)\mathcal{L}_K, \tag{2}$$

*where the inequalities are to be understood in the sense of the semidefinite matrix ordering. Equivalently,*

$$(1-\varepsilon)x^T\mathcal{L}_K x \leq x^T\mathcal{L}_J x \leq (1+\varepsilon)x^T\mathcal{L}_K x \qquad \forall x \in \mathbb{R}^{n_{k-1}}.$$

A proof of Theorem 3.1 is detailed in Section 3.2.

### 3.2 Proof of Theorem 3.1

In this section, we provide a detailed proof of theorem 3.1. Our proof closely follows the proof of Spielman and Srivastava [63, Theorem 1].

Following the definitions from Section 3.1, let $\mathcal{L} = W_{k-1}\mathcal{L}_K$ and $\tilde{\mathcal{L}} = W_{k-1}\mathcal{L}_J$. $\mathcal{L}$ and $\tilde{\mathcal{L}}$ are symmetric positive semidefinite matrices. When our sparsification algorithm is applied to sparsify $K$ at dimension $k$, all simplices in $K$ of dimensions up to $k-1$ are simply copied to $J$ along with the corresponding weights so that the weight matrix $W_{k-1}$ is the same for $K$ and $J$. To prove theorem 3.1, we will first establish that

$$(1-\varepsilon)\mathcal{L} \preceq \tilde{\mathcal{L}} \preceq (1+\varepsilon)\mathcal{L},$$

and then show that (2) holds if and only if the inequality above holds.

Since $\mathcal{L}$ is symmetric, positive semidefinite, $\mathcal{L}^+$ and $R_k$ are also symmetric positive semidefinite matrices. We define the matrix $\Pi = W_k^{1/2} R_k W_k^{1/2}$.

**Lemma 3.1.** *The matrix $\Pi = W_k^{1/2} R_k W_k^{1/2}$ defined above has the following properties:*

*(i)* $\Pi$ *is a projection matrix.*

*(ii)* $\mathrm{Im}(\Pi) = \mathrm{Im}(W_k^{1/2} D_{k-1})$.

*(iii)* $\Pi(f,f) = \|\Pi(\cdot, f)\|^2$.

*(iv)* $\mathrm{Rank}(\Pi) = \mathrm{Tr}(\Pi) \leq n_{k-1}$.

*Proof.* (i) Observe that

$$\begin{aligned}
\Pi\Pi &= (W_k^{1/2} D_{k-1} \mathcal{L}^+ D_{k-1}^T W_k^{1/2})(W_k^{1/2} D_{k-1} \mathcal{L}^+ D_{k-1}^T W_k^{1/2}) \\
&= (W_k^{1/2} D_{k-1} \mathcal{L}^+)(D_{k-1}^T W_k^{1/2} W_k^{1/2} D_{k-1})(\mathcal{L}^+ D_{k-1}^T W_k^{1/2}) \\
&= W_k^{1/2} D_{k-1} \mathcal{L}^+ \mathcal{L} \mathcal{L}^+ D_{k-1}^T W_k^{1/2} \qquad \text{since} \quad \mathcal{L} = D_{k-1}^T W_k D_{k-1} \\
&= W_k^{1/2} D_{k-1} \mathcal{L}^+ D_{k-1}^T W_k^{1/2} \\
&= \Pi
\end{aligned}$$

(ii) First, note that $\mathrm{Im}(\Pi) = \mathrm{Im}(W_k^{1/2} D_{k-1} \mathcal{L}^+ D_{k-1}^T W_k^{1/2}) \subseteq \mathrm{Im}(W_k^{1/2} D_{k-1})$. Now, for any vector $y \in \mathrm{Im}(W_k^{1/2} D_{k-1})$, there exists a vector $x \perp \mathrm{Ker}(W_k^{1/2} D_{k-1}) = \mathrm{Ker}(\mathcal{L})$ such that $y = W_k^{1/2} D_{k-1} x$. Then,

$$\begin{aligned}
\Pi y &= (W_k^{1/2} D_{k-1} \mathcal{L}^+ D_{k-1}^T W_k^{1/2})(W_k^{1/2} D_{k-1} x) \\
&= (W_k^{1/2} D_{k-1} \mathcal{L}^+)(D_{k-1}^T W_k^{1/2} W_k^{1/2} D_{k-1}) x \\
&= W_k^{1/2} D_{k-1} \mathcal{L}^+ \mathcal{L} x \\
&= W_k^{1/2} D_{k-1} x \\
&= y.
\end{aligned}$$

Therefore, $y \in \mathrm{Im}(\Pi)$.

(iii) We have, $\Pi(f,f) = \Pi^2(f,f)$, and since $\Pi$ is symmetric,

$$\Pi^2(f,f) = \Pi(\cdot, f)^T \Pi(\cdot, f) = \|\Pi(\cdot, f)\|^2 .$$

(iv) Since $\Pi^2 = \Pi$, all eigenvalues of $\Pi$ are either 0 or 1. Therefore, $\mathrm{Rank}(\Pi) = \mathrm{Tr}(\Pi)$. And since $D_{k-1}$ is an $n_k \times n_{k-1}$ matrix,

$$\dim(\mathrm{Im}(\Pi)) = \dim(\mathrm{Im}(W_k^{1/2} D_{k-1})) \leq n_{k-1}.$$

□

We also define the $n_k \times n_k$ non-negative, diagonal matrix $Q_k$ with entries:

$$Q_k(f,f) = \frac{\tilde{w}_f}{w_f} = \frac{\#\text{ times } f \text{ is sampled}}{qp_f},$$

where the random entry $Q_k(f,f)$ captures the "amount" of $k$-simplex $f$ included in $J$ by **Sparsify**. The weight of simplex $f$ in $J$ is $\tilde{w}_f = Q_k(f,f)w_f$. The weight matrix can be written as $\tilde{W}_k = W_k Q_k = W_k^{1/2} Q_k W_k^{1/2}$. The $(k-1)$-up Laplacian $\tilde{\mathcal{L}}$ of the sparse complex can be written as

$$\tilde{\mathcal{L}} = D_{k-1}^T \tilde{W}_k D_{k-1} = D_{k-1}^T (W_k^{1/2} Q_k W_k^{1/2}) D_{k-1}.$$

The scaling by $1/qp_f$ in $Q_k$ ensures that $\mathbb{E}\tilde{W}_k = W_k$. As a result, we have $\mathbb{E}Q_k = I$ and $\mathbb{E}\tilde{\mathcal{L}} = \mathcal{L}$.

**Lemma 3.2** (Rudelson and Vershynin [60]). *Let* $\mathbf{p}$ *be a probability distribution over* $\Omega \subseteq \mathbb{R}^d$ *such that* $\sup_{y\in\Omega} \|y\|_2 \le M$ *and* $\left\|\mathbb{E}_{\mathbf{p}} yy^T\right\|_2 \le 1$. *Let* $y_1, y_2, \ldots, y_q$ *be independent samples drawn from* $\mathbf{p}$. *Then*

$$\mathbb{E}\left\| \frac{1}{q}\sum_{i=1}^{q} y_i y_i^T - \mathbb{E}yy^T \right\|_2 \le \min\left\{ CM\sqrt{\frac{\log(q)}{q}},\ 1 \right\},$$

*where* $C$ *is an absolute constant.*

The matrix $\Pi Q_k \Pi$ can be expressed as the average of symmetric rank one matrices:

$$\begin{aligned}
\Pi Q_k \Pi &= \sum_f Q_k(f,f)\Pi(\cdot,f)\Pi(\cdot,f)^T \\
&= \sum_f \frac{(\#\text{ times } f \text{ is sampled})}{qp_f}\Pi(\cdot,f)\Pi(\cdot,f)^T \\
&= \frac{1}{q}\sum_f (\#\text{ times } f \text{ is sampled})\frac{\Pi(\cdot,f)}{\sqrt{p_f}}\frac{\Pi(\cdot,f)^T}{\sqrt{p_f}} \\
&= \frac{1}{q}\sum_{i=1}^{q} y_i y_i^T.
\end{aligned}$$

Vectors $y_i$ are drawn independently with replacement from the distribution

$$y = \frac{1}{\sqrt{p_f}}\Pi(\cdot,f) \quad \text{with probability } p_f.$$

The expectation of $yy^T$ is given by

$$\mathbb{E}yy^T = \sum_f p_f \frac{\Pi(\cdot,f)}{\sqrt{p_f}}\frac{\Pi(\cdot,f)^T}{\sqrt{p_f}} = \Pi\Pi = \Pi.$$

Therefore, $\left\|\mathbb{E}yy^T\right\|_2 = \|\Pi\|_2 = 1$. A bound on the norm of $y$ is given by

$$\frac{1}{\sqrt{p_f}}\|\Pi(\cdot,f)\|_2 = \frac{\sqrt{\Pi(f,f)}}{\sqrt{p_f}} = \sqrt{\sum_f w_f R_k(f,f)} = \sqrt{\mathrm{Tr}(\Pi)} \leq n_{k-1}.$$

Now, using lemma 3.2, with $q = 9C^2 n_{k-1}\ln n_{k-1}/\varepsilon^2$, we have

$$\mathbb{E}\,\|\Pi Q_k \Pi\|_2 = \mathbb{E}\left\|\frac{1}{q}\sum_{i=1}^{q} y_i y_i^T\right\|_2 \leq C\sqrt{\varepsilon^2 n_{k-1}\frac{\ln\left(9C^2 n_{k-1}\ln n_{k-1}/\varepsilon^2\right)}{9C^2 n_{k-1}\ln n_{k-1}}} \leq \frac{\varepsilon}{2},$$

for sufficiently large $n_{k-1}$, as $\varepsilon$ is assumed to be at least $1/\sqrt{n_{k-1}}$. Then, by Markov's inequality, we have $\|\Pi Q_k \Pi - \Pi\Pi\|_2 \leq \varepsilon$ with a probability at least $1/2$.

**Lemma 3.3.** *Suppose $Q_k$ is non-negative diagonal matrix such that $\|\Pi Q_k \Pi - \Pi\Pi\|_2 \leq \varepsilon$. Then, for all $x \in \mathbb{R}^{n_{k-1}}$,*

$$(1-\varepsilon)x^T\mathcal{L}x \leq x^T\tilde{\mathcal{L}}x \leq (1+\varepsilon)x^T\mathcal{L}x$$

*where $\mathcal{L} = D_{k-1}^T W_k D_{k-1}$ and $\tilde{\mathcal{L}} = D_{k-1}^T W_k^{1/2} Q_k W_k^{1/2} D_{k-1}$.*

*Proof.* For a symmetric matrix $A$, $\|A\|_2 = \sup_{y\neq 0}\frac{|y^TAy|}{y^Ty}$. Therefore, the assumption that $\|\Pi Q_k \Pi - \Pi\Pi\|_2 \leq \varepsilon$ is equivalent to

$$\sup_{y\in\mathbb{R}^{n_k}, y\neq 0}\frac{|y^T\Pi(Q_k - I)\Pi y|}{y^Ty} \leq \varepsilon.$$

Note that if $x \in \ker(W_k^{1/2}D_{k-1})$, $x^T\mathcal{L}x = x^T\tilde{\mathcal{L}}x = 0$ and the claim holds trivially. If $x \notin \ker(W_k^{1/2}D_{k-1})$, then a vector $y = W_k^{1/2}D_{k-1}x$ is in $\mathrm{Im}(W_k^{1/2}D_{k-1})$. Restricting our attention to such vectors, we have

$$\sup_{y\in\mathrm{Im}(W_k^{1/2}D_{k-1}), y\neq 0}\frac{|y^T\Pi(Q_k - I)\Pi y|}{y^Ty} \leq \varepsilon.$$

However, from lemma 1.2 we have $\Pi y = y$ for any $y \in \mathrm{Im}(W_k^{1/2}D_{k-1})$. Therefore, we have,

$$\begin{aligned}
&\sup_{y\in\mathrm{Im}(W_k^{1/2}D_{k-1}), y\neq 0}\frac{|y^T\Pi(Q_k - I)\Pi y|}{y^Ty}\\
&= \sup_{y\in\mathrm{Im}(W_k^{1/2}D_{k-1}), y\neq 0}\frac{|y^T(Q_k - I)y|}{y^Ty}\\
&= \sup_{x\in\mathbb{R}^{n_{k-1}}, W_k^{1/2}D_{k-1}x\neq 0}\frac{|x^TD_{k-1}^TW_k^{1/2}(Q_k - I)W_k^{1/2}D_{k-1}x|}{x^TD_{k-1}^TW_kD_{k-1}x}\\
&= \sup_{x\in\mathbb{R}^{n_{k-1}}, W_k^{1/2}D_{k-1}x\neq 0}\frac{|x^TD_{k-1}^TW_k^{1/2}Q_kW_k^{1/2}D_{k-1}x - x^TD_{k-1}^TW_kD_{k-1}x|}{x^TD_{k-1}^TW_kD_{k-1}x}\\
&= \sup_{x\in\mathbb{R}^{n_{k-1}}, W_k^{1/2}D_{k-1}x\neq 0}\frac{|x^T\tilde{\mathcal{L}}x - x^T\mathcal{L}x|}{x^T\mathcal{L}x}.
\end{aligned}$$

Therefore, if $\|\Pi Q_k \Pi - \Pi\Pi\|_2 \leq \varepsilon$, then

$$\sup_{x \in \mathbb{R}^{n_{k-1}}, W_k^{1/2} D_{k-1} x \neq 0} \frac{|x^T \tilde{\mathcal{L}} x - x^T \mathcal{L} x|}{x^T \mathcal{L} x} \leq \varepsilon.$$

Rearranging the terms, we get for all $x \in \mathbb{R}^{n_{k-1}}$,

$$(1-\varepsilon) x^T \mathcal{L} x \leq x^T \tilde{\mathcal{L}} x \leq (1+\varepsilon) x^T \mathcal{L} x,$$

which is equivalent to

$$(1-\varepsilon)\mathcal{L} \preceq \tilde{\mathcal{L}} \preceq (1+\varepsilon)\mathcal{L}. \tag{3}$$

□

Now, to show that (2) holds, we use the following elementary lemma (whose proof is included for completeness):

**Lemma 3.4.** *For any symmetric positive semidefinite matrices $A$ and $B$ and any positive definite diagonal matrix $D$, we have $A \succeq B$ if and only if $DA \succeq DB$.*

*Proof.* First, assume $A \succeq B$. Let $C = A - B$. Then, $C \succeq 0$. Since $D$ is a positive definite diagonal matrix, $D^{1/2} C D^{1/2} \succeq 0$. However, $D^{1/2} C D^{1/2}$ is similar to $DC$ because

$$D^{1/2} C D^{1/2} = D^{-1/2} (DC) D^{1/2}.$$

Therefore, $DC$ has the same eigenvalues as $D^{1/2} C D^{1/2}$ which means $DC \succeq 0$ or equivalently, $DA \succeq DB$.

Now suppose $DC \succeq 0$. Then, $D^{1/2} C D^{1/2} \succeq 0$, due to similarity. However,

$$C = D^{-1/2} (D^{1/2} C D^{1/2}) D^{-1/2},$$

and therefore $C \succeq 0$ or equivalently, $A \succeq B$. □

We can write the up Laplacians as $\mathcal{L}_K = W_{k-1}^{-1} \mathcal{L}$ and $\mathcal{L}_J = W_{k-1}^{-1} \tilde{\mathcal{L}}$. Since $W_{k-1}$ is a diagonal matrix of positive weights, $W_{k-1}^{-1}$ is a positive definite diagonal matrix. Therefore, according to lemma 3.4, if inequality (3) holds, inequality (2) must hold, i.e.,

$$(1-\varepsilon)\mathcal{L}_K \preceq \mathcal{L}_J \preceq (1+\varepsilon)\mathcal{L}_K.$$

**Corollary 3.1.** *Suppose $Z_f$ are numbers satisfying $Z_f \geq R_k(f,f)/\alpha$, and $\sum_f w_f Z_f \leq \alpha \sum_f w_f R_k(f,f)$ for some $\alpha \geq 1$. If we sample as in* **Sparsify** *but take each $k$-simplex $f$ with probability $p'_f = w_f Z_f / \sum_f w_f Z_f$ instead of $p_f = w_f R_k(f,f)/\sum_f w_f R_k(f,f)$, then the resulting sparse complex $J$ satisfies*

$$(1-\varepsilon\alpha)\mathcal{L} \preceq \tilde{\mathcal{L}} \preceq (1+\varepsilon\alpha)\mathcal{L}.$$

*Proof.* Note that

$$p'_f = \frac{w_f Z_f}{\sum_f w_f Z_f} \geq \frac{w_f R_k(f,f)\alpha}{\alpha \sum_f w_f R_k(f,f)} = \frac{p_f}{\alpha^2}.$$

Proceeding as in the proof for theorem 3.1, the new bound on the norm of random vector $y$ is given by

$$\frac{1}{\sqrt{p'_f}} \|\Pi(\cdot, f)\|_2 \leq \frac{\alpha}{\sqrt{p_f}} \sqrt{\Pi(f,f)} = \alpha\sqrt{\mathrm{Tr}(\Pi)}.$$

Thus, constant factor approximation of generalized effective resistances introduces the same constant factor, $\alpha$, in the bound on expectation in lemma 3.2, and consequently in the final inequality, but does not change anything else. □

## 4 Generalized Cheeger inequalities for simplicial complexes

In Section 4.1, we show that the Cheeger constant of the sparsified simplicial complex is bounded below by a multiplicative factor of the first nontrivial eigenvalue of the up Laplacian for the original complex (via Theorem 4.2 and Corollary 4.1). We give the proof of Theorem 4.2 in Section 4.2.

### 4.1 Cheeger constant of sparsified simplicial complexes

**Cheeger constant and inequality for graphs.** We begin with the following definition of the Cheeger constant for an unweighted graph $G = (V, E)$ used by Gundert and Szedlák [35]:

$$h(G) := \min_{\varnothing \subsetneq A \subsetneq V} \frac{|V|\ |E(A, V \setminus A)|}{|A|\ |V \setminus A|}, \tag{4}$$

where $E(A, V \setminus A)$ is the set of edges that connect $A \subset V$ to $(V \setminus A) \subset V$. For a weighted graph, $G = (V, E, w)$, this definition is typically generalized to

$$h(G) := \min_{\varnothing \subsetneq A \subsetneq V} \frac{|V|}{|A|\ |V \setminus A|} \sum_{(i,j) \in E(A, V \setminus A)} w_{ij}. \tag{5}$$

The Cheeger inequality for graphs takes the form $c \cdot \lambda_1(L_G) \leq h(G) \leq C \cdot \sqrt{\lambda_1(L_G)}$, where $\lambda_1$ is the first nontrivial eigenvalue of a graph Laplacian. The constants $c$ and $C$ depend on the choice of definition for the Cheeger constant and the graph Laplaican; see, e.g., [15, Chapter 2]. Using the variational formulation for eigenvalues and a suitable test function, we have no difficulty proving that for the weighted (un-normlized) graph Laplacian, the lower bound for the Cheeger constant defined in (5) is given by $\frac{1}{2} \cdot \lambda_1(L_G) \leq h(G)$. Here, we prove an analogous inequality for weighted simplicial complexes, which we refer to as the generalized Cheeger inequality. This inequality gives a lower bound on the Cheeger constant; an upper bound is not possible for weighted simplicial complexes by the argument of Gundert and Szedlák [35, p.5].

**Generalized Cheeger inequality for simplicial complexes of Gundert and Szedlák.** We first recall the generalized Cheeger inequality for simplicial complexes of Gundert and

Szedlák [35]. For a $k$-dimensional simplicial complex $K$, its *$k$-dimensional completion* is defined to be

$$\bar{K} := K \bigcup \left\{ \tau^* \in \binom{V}{k+1} | (\tau^* \backslash \{v\}) \in X, \forall v \in \tau^* \right\}.$$

$\bar{K}$ is the complete $k$-dimensional complex when $K$ has a complete $(k-1)$-skeleton. The generalized Cheeger constant for *unweighted* simplicial complexes is defined to be

$$h(K) := \min_{\substack{V = \bigsqcup_{i=0}^{k} A_i \\ A_i \neq \varnothing}} \frac{|V||F(A_0, A_1, \ldots, A_k)|}{|F^*(A_0, A_1, \ldots, A_k)|}, \tag{6}$$

where $F(A_0, A_1, \ldots, A_k)$ and $F^*(A_0, A_1, \ldots, A_k)$ are the sets of all $k$-simplices of $K$ and $\overline{K}$, respectively, with one vertex in $A_i$ for all $0 \leq i \leq k$.

**Theorem 4.1** ([35, Theorem 2])**.** *If $\lambda_1(\mathcal{L}_K)$ is the first nontrivial eigenvalue of the $k$-th up Laplacian and if every $(k-1)$-face is contained in at most $C^*$ $k$-face of $K$, then*

$$\frac{|V|}{(k+1)\ C^*} \cdot \lambda_1(\mathcal{L}_K) \leq h(K).$$

**Remark.** Recall that the Cheeger inequality for graphs includes an upper bound of the Cheeger constant $h(G)$ in terms of $\lambda_1(L_G)$. However, as pointed out by Gundert and Szedlák, $\lambda_1(\mathcal{L}_K) = 0$ does not imply $h(K) = 0$ [35]. Therefore, a higher dimensional analogue of this upper bound of the form $h(K) \leq C \cdot \lambda_1(\mathcal{L}_K)^{\frac{1}{m}}$ is not possible. We also remark that an alternative Cheeger inequality is given in [57].

**A generalized Cheeger constant for weighted simplicial complexes.** Analogous to the generalization of the unweighted Cheeger constant in Equation (4) to the weighted Cheeger constant in Equation (5), we define the generalized Cheeger constant for weighted simplicial complexes by

$$h(K) := \min_{\substack{V = \bigsqcup_{i=0}^{k} A_i \\ A_i \neq \varnothing}} \frac{|V|}{|F^*(A_0, A_1, \ldots, A_k)|} \sum_{X \in F(A_0, A_1, \ldots, A_k)} w_k(X). \tag{7}$$

Observe that Equation (7) agrees with Equation (6) in the case when all weights are unity. The following result can be proved analogously to Theorem 4.1:

**Theorem 4.2.** *Let $\lambda_1(\mathcal{L}_K)$ be the first nontrivial eigenvalue of the $(k-1)$-th weighted up Laplacian $\mathcal{L}_K$. If every $(k-1)$-face $\sigma$ is contained in at most $C^*$ $k$-faces of $\bar{K}$ and $w_{k-1}(\sigma) \geq W^* > 0$, then*

$$\frac{|V|\ W^*}{(k+1)\ C^*} \cdot \lambda_1(\mathcal{L}_K) \leq h(K).$$

A proof of Theorem 4.2 is given in Section 4.2. Combining Theorem 3.1 and Theorem 4.2 leads to the following result:

**Corollary 4.1.** *In the setting as Theorem 3.1 and Theorem 4.2, we have with probability* $\frac{1}{2}$

$$\frac{|V|\ W^*}{(k+1)\ C^*}(1-\varepsilon)\cdot\lambda_1(\mathcal{L}_K) \leq \frac{|V|\ W^*}{(k+1)\ C^*}\cdot\lambda_1(\mathcal{L}_J) \leq h(J).$$

Thus, the Cheeger constant of the sparsified simplicial complex, $J$, is bounded below by a multiplicative factor of the first nontrivial eigenvalue of the up Laplacian for the original complex, $K$.

### 4.2 Proof of Theorem 4.2

Note that $\mathrm{Ker}(\mathcal{L}_K) = \mathrm{Ker}(\delta_{k-1}) = Z^{k-1}$. Since $B^{k-1} \subseteq Z^{k-1}$, the eigenvectors corresponding to the nonzero eigenvalues of $\mathcal{L}_K$ are contained in $(B^{k-1})^\perp$. By the Hodge decomposition, we know that $(B^{k-1})^\perp = Z_{k-1}$. Therefore, $\lambda_1(\mathcal{L}_K)$ can be formulated as [35, Section 4, Equation (2)]

$$\lambda_1(\mathcal{L}_K) = \min_{f\in Z_{k-1}} \frac{(\mathcal{L}_K f, f)_{C^{k-1}}}{(f,f)_{C^{k-1}}}, \tag{8}$$

where $(a,b)_{C^k} = \sum_{\sigma\in S_k} w_k(\sigma)a(\sigma)b(\sigma)$ for all $a, b \in C^k$ is the inner product defined over $C^k$, the space of all real valued functions on $S_k$. We will omit the subscript $C^k$ from here on. The key idea in the proof is to find a function $f \in Z_{k-1}$ such that

$$\frac{(\mathcal{L}_K f, f)}{(f,f)} = h(K).$$

In order to define such a function, we fix a partition $A_0, \ldots, A_k$ of the vertex set $V$ of $K$, which realizes the minimum in equation 7. We will refer to $A_i$'s as blocks. For simplicity, we choose a linear ordering on $V$ such that for all $w \in A_i$ and $v \in A_j$ we have $w < v$ if $i < j$. To keep the notation simple, we will simply write $F$ and $F^*$ instead of $F(A_0,\ldots,A_k)$ and $F^*(A_0,\ldots,A_k)$. Note that $\lambda_1(\mathcal{L}_K)$ does not depend on the choice of orientation.

Let $\sigma = [v_0, \ldots, v_{k-1}] \in S_{k-1}$. Then $f \in C^{k-1}$ is defined as

$$f(\sigma) = \begin{cases} (-1)^l |A_l| & \text{if } A_l \text{ is the \textit{unique} block not containing any } v_i \\ 0 & \text{otherwise.} \end{cases}$$

**Lemma 4.1.** *Let* $\mathcal{L}_K$ *be the* $(k-1)$*-th weighted up Laplacian of* $K$ *and let* $f$ *be defined as above. Then,*

$$(\mathcal{L}_K f, f) = (\delta_{k-1} f, \delta_{k-1} f) = |V|^2 \sum_{\tau\in F} w_k(\tau).$$

*Proof.* Consider $\tau = [v_0, \ldots, v_k] \in S_k$. If $\tau \in F$, i.e., if $v_i \in A_i$ for all $i = 0, \ldots, k$, then

$$(\delta_{k-1})f(\tau) = \sum_{i=0}^{k} [\tau : \tau\backslash\{v_i\}] f(\tau\backslash\{v_i\}) = \sum_{i=0}^{k} (-1)^i (-1)^i |A_i| = |V|.$$

Now suppose $\tau \notin F$, but $v_i, v_j$ is the only pair of vertices in the same block. Let $v_i < v_j$. Then by our chosen linear ordering, $i+1 = j$. If $l$ is not equal to $i$ or $i+1$, then $f(\tau\backslash\{v_l\}) = 0$.

The only nonzero terms are $f(\tau\backslash\{v_i\}) = f(\tau\backslash\{v_j\})$. However, these terms cancel out because $[\tau\colon \tau\backslash\{v_i\}] = -[\tau\colon \tau\backslash\{v_{i+1}\}]$.

If three vertices belong to the same block or if two pairs of vertices belong to the same blocks, or indeed in any other arrangement not covered before, then we have at least two empty blocks and $f$ is zero. Therefore,

$$(\delta_{k-1}f)(\tau) = \begin{cases} |V| & \text{if } \tau \in F, \\ 0 & \text{otherwise,} \end{cases}$$

and,

$$(\delta_{k-1}f, \delta_{k-1}f) = \sum_{\tau\in F} w_k(\tau)((\delta_{k-1}f)(\tau))^2 = |V|^2 \sum_{\tau\in F} w_k(\tau).$$

□

**Lemma 4.2.** *Let $f \in C^{k-1}$ be as previously defined. Then, unique $z \in Z_{k-1}$, $b \in B^{k-1}$ such that $f = z + b$ exist. Furthermore,*

$$\lambda_1(\mathcal{L}_K) \leq \frac{|V|^2}{(z,z)} \sum_{\tau\in F} w_k(\tau).$$

*Proof.* Since $Z_{k-1} = (B^{k-1})^\perp$, unique cochains $z \in Z_{k-1}$ and $b \in B^{k-1}$ such that $f = z + b$ exist. Also, $(\mathcal{L}_K z, z) = (\mathcal{L}_K f, f)$, because $b \in B^{k-1} \subseteq \mathrm{Ker}(\mathcal{L}_K)$. The claim now follows from this fact and using equation 8 and lemma 4.1. □

**Lemma 4.3.** *Let $f \in C^{k-1}$ be as previously defined and let $g \in C^{k-2}$ be arbitrary. For $\tau^* \in F^*$, define*

$$q(\tau^*, g) := \sum_{\sigma\subseteq\tau^*, \sigma\in S_{k-1}} \frac{w_{k-1}(\sigma)}{d(\sigma)} (f(\sigma) - \delta_{k-2}g(\sigma))^2,$$

*where for all $\sigma \in S_{k-1}$, $d(\sigma) = |\{\tau^* \supseteq \sigma | \tau^* \in F^*\}|$. Then,*

1. $(f - \delta_{k-2}g, f - \delta_{k-2}g) \geq \sum_{\tau^*\in F^*} q(\tau^*, g)$.
2. *For $\tau^* = \{v_0, v_1, \ldots, v_k\} \in F^*$ with $v_0 < v_1 < \cdots < v_k$,*

$$q(\tau^*, g) \geq \frac{|V|^2}{\sum_{i=0}^{k} \frac{d(\tau\backslash\{v_i\})}{w_{k-1}(\tau\backslash\{v_i\})}}.$$

*Proof.* 1. By definition of the inner product,

$$(f - \delta_{k-2}g, f - \delta_{k-2}g) = \sum_{\sigma\in S_{k-1}} w_{k-1}(\sigma)(f(\sigma) - \delta_{k-2}g(\sigma))^2.$$

Now, consider the sum on the right-hand side

$$\sum_{\tau^*\in F^*} q(\tau^*, g) = \sum_{\tau^*\in F^*} \sum_{\sigma\subseteq\tau^*, \sigma\in S_{k-1}} \frac{w_{k-1}(\sigma)}{d(\sigma)} (f(\sigma) - \delta_{k-2}g(\sigma))^2.$$

Note that for any $\sigma \in S_{k-1}$ such that $\sigma \subseteq \tau^*$, the corresponding term in the summation appears exactly $d(\sigma)$ times. If $\sigma \not\subseteq \tau^*$, then the corresponding term does not appear at all. Therefore,

$$\sum_{\tau^* \in F^*} q(\tau^*, g) \leq \sum_{\sigma \in S_{k-1}} w_{k-1}(\sigma)(f(\sigma) - \delta_{k-2} g(\sigma))^2.$$

2. Let $\tau^* = [v_0, \ldots, v_k] \in F^*$, such that $v_i \in A_i$ for $i = 1, \ldots, k$. Then,

$$\begin{aligned} q(\tau^*, g) &= \sum_{i=0}^{k} \frac{w_{k-1}(\tau^* \backslash \{v_i\})}{d(\tau^* \backslash \{v_i\})}((-1)^i |A_i| - \delta_{k-2} g(\tau^* \backslash \{v_i\}))^2 \\ &= \sum_{i=0}^{k} \frac{w_{k-1}(\tau^* \backslash \{v_i\})}{d(\tau^* \backslash \{v_i\})}(|A_i| - [\tau^* \colon \tau^* \backslash \{v_i\}] \delta_{k-2} g(\tau^* \backslash \{v_i\}))^2. \end{aligned}$$

Note that the *oriented incidence number* $[\tau^* \colon \sigma]$ is $(-1)^i$ if $\sigma = \tau^* \backslash \{v_i\}$ for $i = 1, \ldots, k$ and 0 if $\sigma \not\subseteq \tau^*$. We also observe that $\sum_{i=0}^{k} [\tau^* \colon \tau^* \backslash \{v_i\}] \delta_{k-2} g(\tau^* \backslash \{v_i\}) = \delta_{k-1} \delta_{k-2} g(\tau^* \backslash \{v_i\}) = 0$. Therefore,

$$q(\tau^*, g) = \sum_{i=0}^{k} \frac{w_{k-1}(\tau^* \backslash \{v_i\})}{d(\tau^* \backslash \{v_i\})} |A_i|^2.$$

Now, using the following version of Cauchy-Schwarz inequality (Titu's lemma / Engel's form) for positive real numbers,

$$\sum_{i=0}^{k} \frac{a_i^2}{b_i} \geq \frac{(\sum_{i=0}^{k} a_i)^2}{\sum_{i=0}^{k} b_i},$$

we obtain

$$q(\tau^*, g) \geq \frac{(\sum_{i=0}^{k} |A_i|)^2}{\sum_{i=0}^{k} \frac{d(\tau \backslash \{v_i\})}{w_{k-1}(\tau \backslash \{v_i\})}} = \frac{|V|^2}{\sum_{i=0}^{k} \frac{d(\tau \backslash \{v_i\})}{w_{k-1}(\tau \backslash \{v_i\})}}.$$

□

Finally, from lemma 4.2, recall that $f = z + b$ where $z \in Z_{k-1}$ and $b \in B^{k-1}$. Therefore, some $g \in C^{k-2}$ such that $f - z = b = \delta_{k-2} g$ exists. By lemma 4.3,

$$(z, z) = (f - \delta_{k-2} g, f - \delta_{k-2} g) \geq \sum_{\tau^* \in F^*} \frac{|V|^2}{\sum_{i=0}^{k} \frac{d(\tau \backslash \{v_i\})}{w_{k-1}(\tau \backslash \{v_i\})}}.$$

Now, if every $(k-1)$-face $\sigma$ of $K$ is contained in $C^*$ $k$-faces of $\bar{K}$ and $w_{k-1}(\sigma) \geq W^*$, then

$$\sum_{i=0}^{k} \frac{d(\tau \backslash \{v_i\})}{w_{k-1}(\tau \backslash \{v_i\})} \leq (k+1) \frac{C^*}{W^*},$$

and

$$(z, z) \geq \frac{|V|^2 \; |F^*| \; W^*}{(k+1) \; C^*}.$$

Using this inequality along with lemma 4.2, we can write

$$\lambda_1(\mathcal{L}_K) \leq \frac{|V|^2 \; (k+1) \; C^*}{|V|^2 \; W^* \; |F^*|} \sum_{\tau \in F} w_k(\tau).$$

Recall that we defined the function $f$ by fixing a partition $A_0, \ldots, A_k$ that realizes the minimum from equation 7, which means

$$h(K) = \frac{|V|}{|F^*|} \sum_{\tau \in F} w_k(\tau),$$

and we get the stated lower bound on $h(K)$:

$$\frac{|V| \; W^*}{(k+1) \; C^*} \lambda_1(\mathcal{L}_K) \leq h(K).$$

## 5 Experimental validation

In Section 5.1, we conduct numerical experiments to *illustrate* the inequalities bounding the spectrum of the up Laplacian of the sparsified simplicial complex, proven in Theorem 3.1. In Section 5.2, we extend a well-known graph spectral clustering method to simplicial complexes. We show that the clusters obtained for sparsified simplicial complexes are similar to those of the original simplicial complex. In Section 5.3, we show results for label propagation on simplicial complexes before and after sparsification.

For each section, we also include the analogous results for graphs to serve as a comparison. We present nothing new in the setting of graphs, Laplacian preservation, spectral clustering, and label propagation; graph-based results are included solely for comparative purposes and to help illustrate our results on simplicial complexes in a more familiar context. We also would like to point out that it is not the focus of this paper to provide the most efficient implementation or to perform large-scale experiments on spectral sparsification of simplicial complexes, but rather, to show that such a sparsification is feasible and theoretically sound. Nevertheless, we have included a discussion on efficient implementations for simplicial complexes in terms of sparsification, spectral clustering, and label propagation in Appendix A. In a nutshell, the naïve implementation of spectral clustering is quadratic in the number of simplices, whereas label propagation is cubic. We can take advantage of sparse matrix methods (see Appendix A for details); our proposed sparsification method could further improve these computational complexity estimates.

### 5.1 Preservation of the spectrum of the up Laplacian for simplicial complexes

**Experimental setup.** Our experimental setup for sparsifying simplicial complexes is an extension of that for graphs; therefore, we begin with a review of graph sparsification. In the setting of graph sparsification [63], we recall that if a graph $H$ is an $\varepsilon$-approximation of a graph $G$ and $n$ is the number of vertices in $H$ and $G$, then we have the following inequality:

$$(1-\varepsilon)x^T L_G x \;\leq\; x^T L_H x \;\leq\; (1+\varepsilon)x^T L_G x, \qquad\qquad \forall x \in \mathbb{R}^n. \tag{9}$$

Subtracting $x^T L_G x$ from all terms in this inequality, we obtain

$$-\varepsilon x^T L_G x \;\leq\; x^T(L_H - L_G)x \;\leq\; \varepsilon x^T(L_G)x, \qquad\qquad \forall x \in \mathbb{R}^n. \tag{10}$$

Let $\lambda_{max}(L_G)$, $\lambda_{max}(L_H)$ and $\lambda_{max}(L_H - L_G)$ be the maximum eigenvalues of $L_G$ and $L_H$ and $L_H - L_G$, respectively. Also, let $\lambda_{min}(L_G)$ be the minimum eigenvalue of $L_G$. Looking at the inequality on the right-hand side of (10), after some algebraic manipulations, we obtain

$$\lambda_{max}(L_H - L_G) = \max_{||x||=1} x^T(L_H - L_G)x \leq \varepsilon \max_{||x||=1} x^T(L_G)x = \varepsilon\lambda_{max}(L_G).$$

Similarly, for the inequality on the left-hand side of (10), we obtain

$$\begin{aligned} 0 = -\varepsilon\lambda_{min}(L_G) = -\varepsilon \min_{||x||=1} x^T L_G x = \max_{||x||=1} -\varepsilon x^T L_G x& \\ \leq \max_{||x||=1} x^T(L_H - L_G)x = \lambda_{max}(L_H - L_G)&. \end{aligned}$$

Together, we have the inequality

$$0 \;\leq\; \lambda_{max}(L_H - L_G) \;\leq\; \varepsilon\lambda_{max}(L_G). \tag{11}$$

Moving from graphs to simplicial complexes, we can obtain the analogous inequality in the setting of simplicial complex sparsification. Let $J$ be a sparsified version of $K$ following the setting of Theorem 3.1. Suppose for a fixed dimension $k$ (where $1 \leq i \leq \dim(K)$), $K$ and $J$ have $(k-1)$-th up Laplacians $\mathcal{L}_K := \mathcal{L}_{K,k-1}$ and $\mathcal{L}_J := \mathcal{L}_{J,k-1}$, respectively, we have

$$(1-\varepsilon)x^T \mathcal{L}_K x \leq x^T \mathcal{L}_J x \leq (1+\varepsilon)x^T \mathcal{L}_K x, \qquad\qquad \forall x \in \mathbb{R}^{n_{k-1}}. \tag{12}$$

A similar argument leads to the following inequality:

$$0 \;\leq\; \lambda_{max}(\mathcal{L}_J - \mathcal{L}_K) \;\leq\; \varepsilon\lambda_{max}(\mathcal{L}_K). \tag{13}$$

Notice that inequality (11) is a special case of the inequality (13).

**Preservation of the spectrum of the sparsified graph Laplacian.** To demonstrate how the spectrum of the graph Laplacian is preserved during graph sparsification, we set up the following experiments. Note that graph sparsification of large graphs is well known; the results described here are used for comparative purposes only. In particular, we would like to give a simple example to compare similar behaviors in preserving the spectrum of up Laplacian for both graphs and simplicial complexes.

Consider a complete graph $G$ with $n_0 = 40$ vertices and $n_1 = 780$ edges. We run multiple sparsification processes on this graph $G$ and study the convergence behavior based on the inequality in (9). For each sparsification process, we use a sequence of sample sizes, ranging between 10 and $2n_1$. For each sample size $q$, we set $\varepsilon = \sqrt{n_0 \log n_0/q}$ by assuming

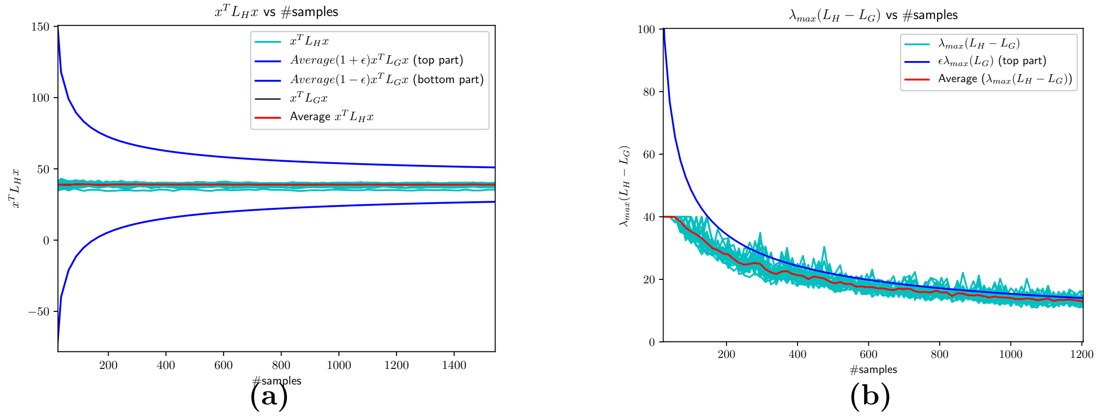


Figure 1: The results of a numerical experiment illustrating inequalities that control the spectrum of sparsified graph Laplacians. **(a)** For an ensemble of vectors, $x \in \mathbb{S}^{n_0}$, and sparsified graphs, $H$, we plot the terms in inequality (9). **(b)** For an ensemble of sparsified graphs, $H$, we plot the terms in the inequality (11).

that $9C^2 = 1$ in the hypothesis of Theorem 3.1. As $q$ varies, we correspondingly obtain a sequence of varying $\varepsilon$ values.

In particular, we run 25 simulations on $G$. For each simulation, we fix a unit vector $x$ uniformly randomly sampled from $\mathbb{S}^{n_0}$ and perform 25 instances of experiments. For each instance, we apply our sparsification procedure to generate the convergence plot using the list of fixed sample sizes $q$ and their corresponding $\varepsilon$'s. Specifically, for each sample size, we obtain a sparse graph $H$ and compute $x^T L_H x$ and $\lambda_{max}(L_H - L_G)$, and we observe the convergence behavior of these quantities as the sample size increases.

In Figure 1(a), we show the convergence behavior based on the inequality in (9). For a single simulation, we compute the point-wise average of $x^T L_H x$ across the 25 instances, and we plot these values as a function of the sample size $q$, which gives rise to a single convergence curve in aqua. Then, we compute the point-wise average of the aqua curves across all simulations, producing the red curve. Since each simulation (for a fixed $x$) has a different upper bound curve $(1 - \varepsilon)x^T L_G x$ and lower bound curve $(1 + \varepsilon)x^T L_G x$, respectively (not shown here), the point-wise average of the upper and lower bound curves across all simulations is plotted in blue. We observe that, on average, these curves reflect the inequality (9), that is, the red curve is nested within its approximated theoretical upper and lower bounds in blue.

In Figure 1(b), we illustrate the theoretical upper and lower bounds for $\lambda_{max}(L_H - L_G)$ given in inequality (11) as the sample size $q$ increases. In particular, we run a single simulation with 25 instances, computing $\lambda_{max}(L_H - L_G)$. Each instance gives us a convergence curve shown in aqua. We compare the point-wise average of $\lambda_{max}(L_H - L_G)$ (in red) with its (approximated) theoretical upper bound in blue and lower bound (i.e., 0, the x-axis). On average, the experimental results respect the inequality (11). Figure 3(a) illustrates how the number of edges increases with the number of samples across all instances.

**Preservation of the spectrum of the up Laplacian for a sparsified simplicial complex.** To demonstrate that the spectrum of the up Laplacian is preserved during the sparsification of a simplicial complex, we set up a similar experiment. We start with a 2-dimensional simplicial complex, $K$, that contains all edges and triangles on $n_0 = 40$ vertices (with $n_1 = 780$ edges and $n_2 = 9880$ faces) and a sequence of fixed sample sizes q. For each sample size $q$, we solve for $\varepsilon = \sqrt{n_1 \log n_1/q}$ assuming that $9C^2 = 1$ in the hypothesis of Theorem 3.1, to get the corresponding sequence of $\varepsilon$ values. With the simplicial complex $K$ and the sequence of sample sizes fixed, we run 25 simulations, each consisting 25 instances and a fixed randomly sampled unit vector $x$ as described previously. This time, however, we sparsify the faces of the simplicial complex by applying Algorithm 1 with $k = 2$. In Figure 2, we plot the terms in inequalities describing the spectrum for these sparsified simplicial complexes.

In Figure 2(a), following the same procedure as for graph sparsification, we obtain a plot that respects the inequality (12). The curves in aqua show the point-wise average of $x^T\mathcal{L}_J x$ across all instances in a single simulation, whereas the red curve represents point-wise average across all instances and all simulations. Since the random vector $x$ is re-sampled for each simulation, the upper and lower bound curves are different for every simulation. In Figure 2(a), we plot their point-wise average across all simulations as the upper and lower bound curves in blue.

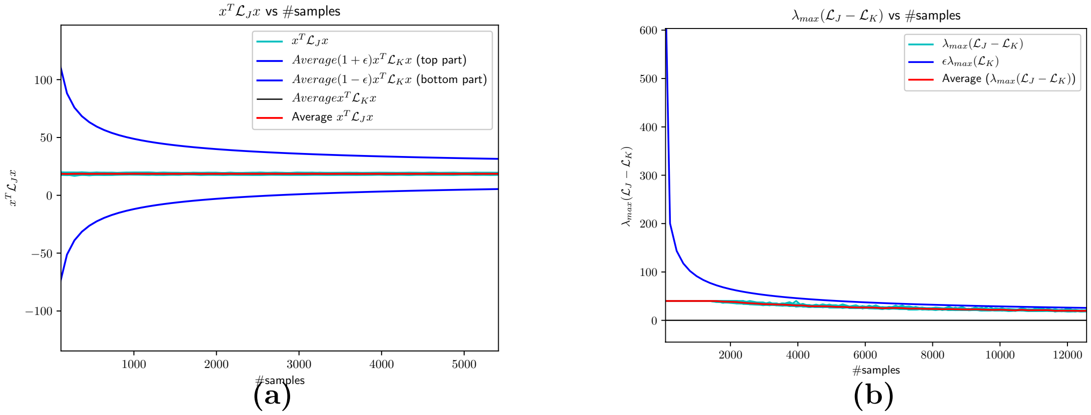
In Figure 2(b), to illustrate inequality (13), we run a single simulation with 25 instances. Each instance gives us a sequence of $\lambda_{max}(\mathcal{L}_J - \mathcal{L}_K)$ values as a function of sample size. We plot them as curves in aqua. We compare the point-wise averages of $\lambda_{max}(\mathcal{L}_J - \mathcal{L}_K)$ (in red) with its (approximated) theoretical upper and lower bounds in blue. Figure 3(b) shows how the number of faces scales with the increasing number of samples across all instances.

### 5.2 Spectral clustering of simplicial complexes

Spectral clustering can be considered as a class of algorithms with many variations. Here, we apply spectral clustering to simplicial complexes before and after sparsification. We demonstrate, via numerical experiments, that preserving the structure of the up Laplacian via sparsification also preserves the results of spectral clustering on simplicial complexes.

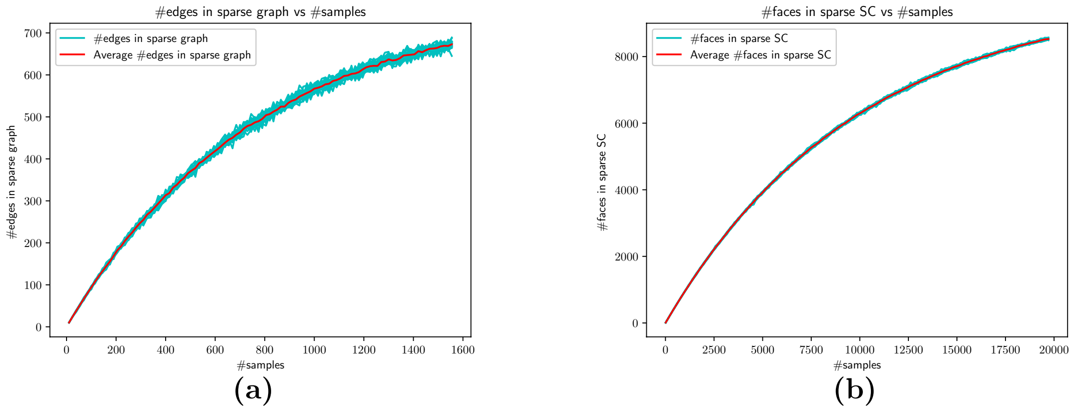
**Datasets.** For comparative purposes, we consider a graph that contains two complete subgraphs, with 20 vertices (and 190 edges) each that are connected by $64 = 8 \times 8$ edges spanning across the two subgraphs. We refer to this graph, $G$, as the *dumbbell graph*; it has $n_0 = 40$ vertices and $n_1 = 444$ edges. All edge weights are set to be 1. To compute the sparsified graph, the number of samples, $q$, is set to be $0.5n_1$.

Similarly, we consider a simplicial complex that contains two complete subcomplexes, with 10 vertices, 45 edges, and 120 triangles each. The two subcomplexes are connected by 16 cross edges and 48 cross triangles so that the simplicial complex is made up of $n_0 = 20$ vertices, $n_1 = 106$ edges, and $n_2 = 288$ triangles. We refer to this simplicial complex, $K$, as the *dumbbell complex*. The weights on all edges and triangles are set to be 1. To compute the sparsified simplicial complex, the number of samples, $q$, is set to be $0.75n_2$. We compare



**(a)** **(b)**

Figure 2: The results of a numerical experiment illustrating inequalities that control the spectrum of the up Laplacian for sparsified simplicial complexes. **(a)** For an ensemble of vectors, $x \in \mathbb{S}^{n_1}$, and sparsified simplicial complexes, $J$, we plot the terms in inequality (12). **(b)** For an ensemble of sparsified simplicial complexes, $J$, we plot the terms in the inequality (13).



**(a)** **(b)**

Figure 3: Plots illustrating how **(a)** the number of edges in the case of graph sparsification and **(b)** the number of faces/triangles in the case of simplicial complex sparsification vary with increasing sample size.

the result of spectral clustering on the dumbbell graph to the result on a dumbbell complex.

**Spectral clustering algorithm for graphs.** We use the Ng-Jordan-Weiss algorithm [50], given here as Algorithm 2, to perform spectral clustering of graphs. Let $n_0$ be the number of vertices in a graph. Recall the *affinity matrix* $A \in \mathbb{R}^{n_0 \times n_0}$ is a matrix where $A(i,j) \geq 0$ captures the affinity (i.e., measure of similarity) between vertex $i$ and vertex $j$. In our setting, $A(i,j)$ corresponds to the weight of edge $e_{ij}$ in the diagonal edge weight matrix $W_1$. The graph Laplacian can be written as $L = \Delta - A$. Furthermore $M = I - L_N$, where

```
Algorithm 2: y = Cluster(G, d)
    Data: A weighted, undirected graph $G$ with $n$ vertices, and the number of
          clusters $d$.
    Result: A vector $\mathbf{y}$ containing cluster assignments $l \in \{1, 2, \ldots, d\}$ for the
            vertices of $G$.
    Construct matrix $A$ where $A(i, j) =$ the weight of edge $e_{ij}$, $A(i, j) = 0$ otherwise.
    Compute diagonal matrix $\Delta \in \mathbb{R}^{n_0 \times n_0}$, where $\Delta(j, j) = \sum_i A(i, j)$.
    $M = \Delta^{-1/2} A \Delta^{-1/2}$.
    Construct matrix $X = [u_1 u_2 \cdots u_d] \in \mathbb{R}^{n \times d}$ where $u_i$'s are the eigenvectors
      corresponding to the $d$ largest eigenvalues of $M$ (chosen to be orthogonal to
      each other in the case of repeated eigenvalues).
    $Y_{ij} = X_{ij} / \left(\sum_j X_{ij}^2\right)^{1/2}$ (normalize rows of $X$ to have unit length).
    $\mathbf{y} = \mathbf{kMeans}(Y, d)$.
    Return $\mathbf{y}$ as cluster assignments for vertices of $G$.
```

$L_N = \Delta^{-1/2} L \Delta^{-1/2}$ is referred to as the normalized graph Laplacian. In the case of a binary graph (where edge weights are either 0 or 1), the affinity matrix $A$ equals the vertex-vertex adjacency matrix, and $\Delta$ is the degree matrix with diagonal elements $\Delta(j, j)$ being the number of edges incident on vertex $v_j$.

To demonstrate the utility of the sparsification, we illustrate the spectral clustering results before and after graph sparsification in Figure 4 (a)-(b). Since graph sparsification preserves the spectral properties of graph Laplacian, we expect it to also preserve (to some extent) the results of spectral methods, such as spectral clustering.

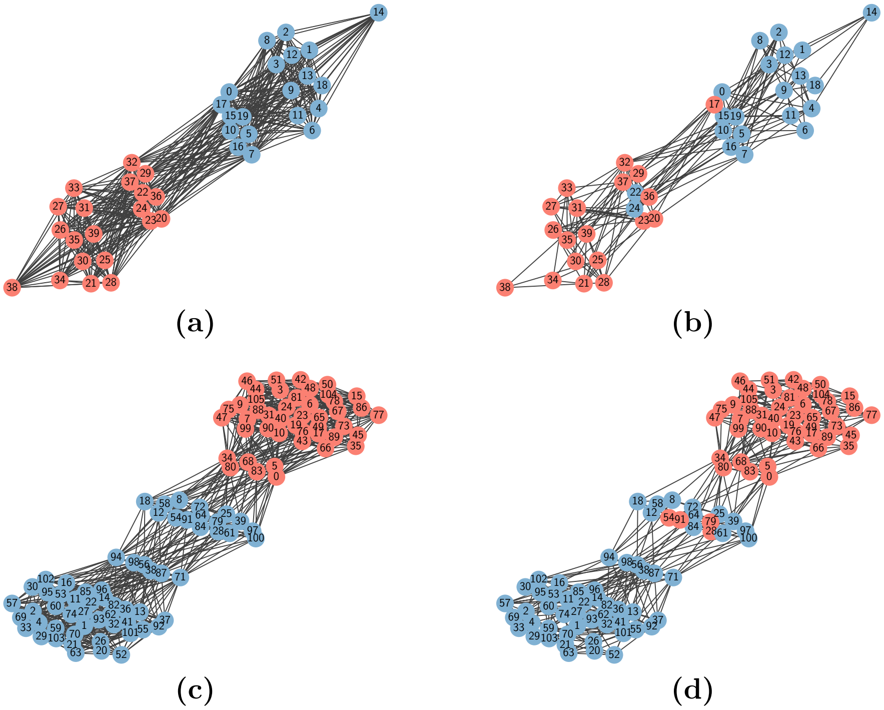
**Spectral clustering algorithm for simplicial complexes.** We seek to extend the Ng-Jordan-Weiss algorithm [50] to simplicial complexes, which has not yet been studied. We seek the simplest generalization by replacing the vertex-vertex affinity matrix with an edge-edge affinity matrix $A_d$, where two edges are considered to be adjacent if they are faces of the same triangle. This definition is a straightforward extension of the adjacency among vertices in graphs; however, it does not account for the orientation of edges or triangles.

Formally, let $n_1$ be the number of edges. We define the edge-edge *affinity matrix* $A_d \in \mathbb{R}^{n_1 \times n_1}$, where

$$A_d(i, j) = \begin{cases} w_f & \text{if } e_i \text{ and } e_j \text{ are both edges of triangle } f \in K \text{ with weight } w_f, \\ 0 & \text{otherwise.} \end{cases}$$

We define $\Delta_d \in \mathbb{R}^{n_1 \times n_1}$ to be the diagonal matrix with element $\Delta_d(j, j)$ being the sum of $A$'s $j$-th column. With $A_d$ and $\Delta_d$ defined this way, we can apply the Ng-Jordan-Weiss algorithm to cluster the edges of the simplicial complex $K$.

This approach is equivalent to applying spectral clustering to the dual graph of $K$. A *dual graph* $G$ of a given simplicial complex $K$ is created as follows: each edge in $K$ becomes a vertex in the dual graph $G$, and there is an edge between two vertices in $G$ if their



(a)

(b)

(c)

(d)

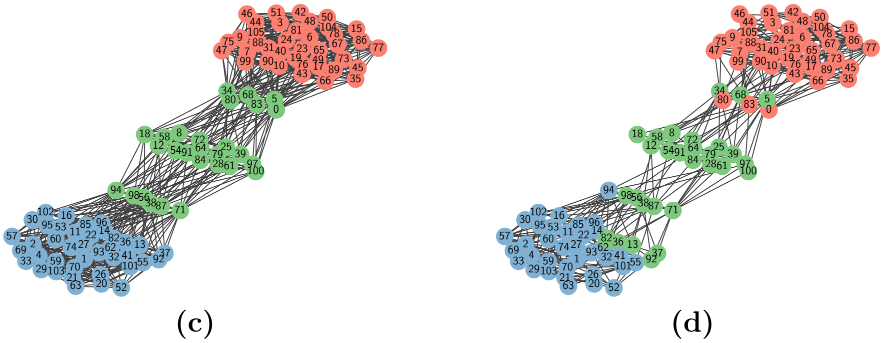
Figure 4: **(a)-(b)**: Spectral clustering of graphs before **(a)** and after **(b)** sparsification. **(c)-(d)**: Spectral clustering of simplicial complexes into two clusters before **(c)** and after **(d)** sparsification. We observe that the clusters are very similar. See Section 5.2 for details.

corresponding edges in $K$ share the same triangle. We then apply spectral clustering to the dual graph $G$ as usual and obtain the resulting clustering of vertices in $G$ (which correspond to the clustering of edges in $K$). To better illustrate our edge clustering results, we visualize the resulting clusters based upon the dual graph. The results are plotted in Figure 4 (c)-(d) for two clusters and Figure 5 for three clusters. Applying the spectral algorithm with these new definitions of $A_d$ and $\Delta_d$ results in clusters that agree reasonably well before and after sparsification.

The affinity matrix, $A_d$, does not take into consideration the orientation of the edges, so the above clustering algorithm does not directly rely on the up Laplacian. One can verify that the dimension 1 up Laplacian can be written as $\mathcal{L}_{K,1} = \Delta_d/2 - A_u$, where $\Delta_d$ is the diagonal matrix defined previously and the *oriented edge-edge affinity matrix*, $A_u \in \mathbb{R}^{n_1 \times n_1}$, is given by

$$A_u(i,j) = \begin{cases} -w_f & \text{edges } e_i \text{ and } e_j \text{ are both faces of the same triangle } f \text{ and both agree or} \\ & \text{disagree with the orientation of their shared triangle,} \\ w_f & \text{if either } e_i \text{ or } e_j \text{ (but not both) agree with the orientation of } f, \\ 0 & \text{if } e_i \text{ and } e_j \text{ are not adjacent.} \end{cases}$$

It follows that $A_d = |A_u|$ where the absolute value operation is applied element-wise. The



(c) (d)

Figure 5: Spectral clustering of simplicial complexes into three clusters before **(a)** and after **(b)** sparsification. See Section 5.2 for details.

relationship between the spectrum of the dual graph Laplacian $L = \Delta_d - A_d$ and the spectrum of up Laplacian, $\mathcal{L}_{K,1}$, used by our sparsification algorithm remains unclear.

### 5.3 Label propagation

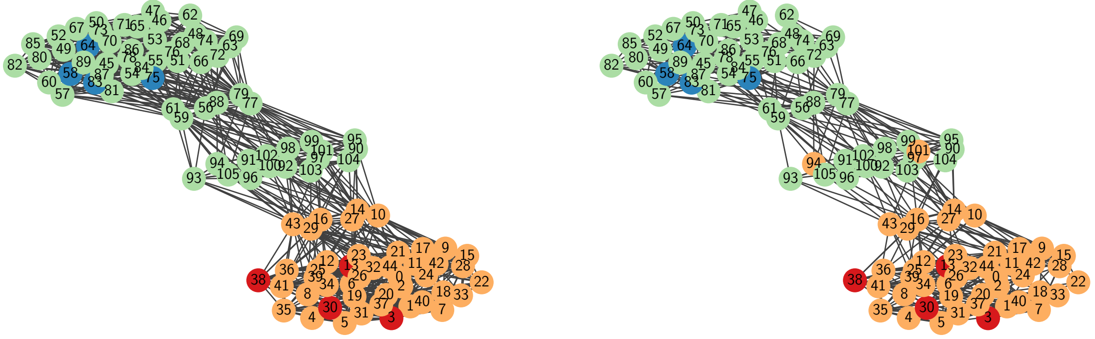
A good example of spectral methods in learning arises from extending label propagation algorithms on graphs to simplicial complexes, in particular, the work by Mukherjee and Steenbergen [48]. Specifically, they adapt the label propagation algorithm to higher dimensional walks on oriented edges, and give visual examples of applying label propagation with the 1-dimensional up Laplacian $\mathcal{L}_1^{up}$, down Laplacian $\mathcal{L}_1^{down}$, and Laplacian $\mathcal{L}_1$. We envision label propagation to be generalized to random walks on even higher dimensional simplices, such as triangles. A direct application of our work is to sparsify the top-dimensional simplices (e.g., triangles in a 2-dimensional simplicial complex) and examine how label propagation behaves on these top-dimensional simplices of the sparsified representation.

Similar to the setting of Section 5.2, we apply and generalize a simple version of label propagation algorithms [75] to the setting of both graphs and simplicial complexes. In particular, as illustrated in Figure 6, we show via the dual graph representation that the results obtained from sparsified simplicial complexes are similar to those of the original simplicial complex. We now describe the algorithmic details.

**Label propagation on graphs.** Again for comparative purposes, we implement a simplified version of the iterative label propagation algorithm [75] based on the notion of stochastic matrix (i.e., random walk matrix) $P = A\Delta^{-1}$, where $A$ is the affinity matrix and $\Delta$ is the diagonal matrix with diagonal elements $\Delta_{jj} = \sum_i A(i,j)$ (as defined in Section 5.2).

The matrix $P$ represents the transition probabilities of the labels. Given $P$ and an initial label distribution $\mathbf{y}$, we iteratively multiply the vector $\mathbf{y}$ by $P$. If the graph is *label-connected* (i.e., we can always reach a labeled vertex from any unlabeled one), then $P^t$ converges to a stationary distribution $x$, that is, $P^t\mathbf{x} = \mathbf{x}$ for a large enough $t$.

Suppose we have two label classes $\{+1, -1\}$. Without loss of generality, assume that first $l$ of the $n$ vertices are assigned labels initially, represented as a length-$l$ vector $\mathbf{y}_l$.



(a) (b)

Figure 6: The results of label propagation on simplicial complexes before **(a)** and after **(b)** sparsification. The red and blue colored vertices correspond to fixed edge labels, and the green and orange colored vertices correspond to propagated edge labels. Blue and green share the same label, and red and orange share the same label. See Section 5.3 for details.

**Algorithm 3: $\mathbf{y} = \textbf{PropagateLabels}(G, \mathbf{y_l})$**

```
Data: A weighted, undirected graph $G$ with $n$ vertices, vector $\mathbf{y_l}$ containing
      labels $\in \{+1, -1\}$ of first $l$ vertices.
Result: A vector $\mathbf{y}$ containing label assignments $l \in \{+1, -1\}$ for all the
        vertices of $G$.
Order the vertices of $G$ so that labels $\mathbf{y_l}$ correspond to the first $l$ vertices.
 Construct matrix $A$ where $A(i,j) =$ the weight of edge $(i,j)$, $A(i,j) = 0$
 otherwise.
Compute diagonal matrix $\Delta \in \mathbb{R}^{n_0 \times n_0}$, where $\Delta(j,j) = \sum_i A_{ij}$.
$P = A\Delta^{-1}$.
Initialize $\mathbf{y}^{(0)} = (\mathbf{y}_l, \mathbf{0})$, $t = 0$.
Repeat until convergence:
    $\mathbf{y}^{(t+1)} = P\mathbf{y}^{(t)}$,
    $\mathbf{y}_l^{(t+1)} = \mathbf{y}_l^{(t)}$.
 Return $\text{sgn}(\mathbf{y}^{(\mathbf{t})})$ as label assignments for vertices of $G$.
```

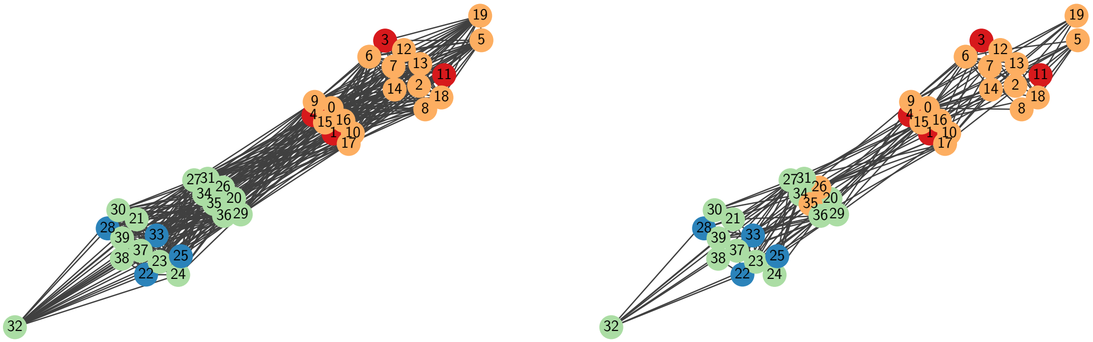
Given a graph $G(V, E)$ and labels $\mathbf{y}_l$, the simplified version of label propagation algorithm is outlined in algorithm 3. Consider $P$ to be divided into blocks as follows:

$$P = \begin{pmatrix} P_{ll} & P_{lu} \\ P_{ul} & P_{uu}, \end{pmatrix}$$

where $l$ and $u$ index the labeled and unlabeled vertices with the number of vertices $n_0 = l + u$. Let $\mathbf{y} = (\mathbf{y}_l, \mathbf{y}_u)$ be the labels at convergence; then $\mathbf{y}_u$ is given by

$$\mathbf{y}_u = (I - P_{uu})^{-1} P_{ul} \mathbf{y_l}$$

As long as our graph is connected, it is also label-connected and $(I - P_{uu})$ nonsingular. Therefore, we can directly compute the labels at convergence without going through the iterative process described in algorithm 3. As illustrated in Figure 7, we apply the label propagation algorithm to the dumbbell graph dataset to demonstrate that preserving the structure of graph Laplacian via sparsification also preserves the results of label propagation on graphs.



**(a)** **(b)**

Figure 7: The results of label propagation on the dumbbell graph before **(a)** and after **(b)** sparsification. The red and blue color represent the initial opposite vertex labels, and the green and orange color correspond to the final propagated vertex labels. Blue and green share the same label, and red and orange share the same label.

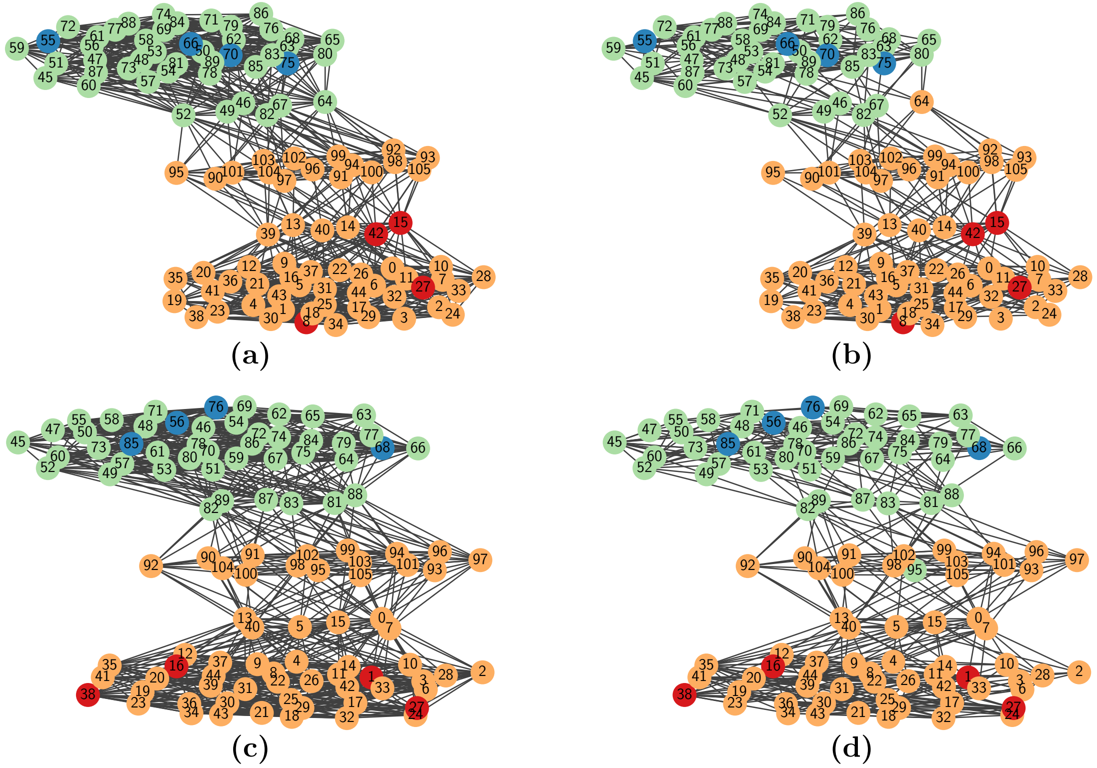
**Label propagation on simplicial complexes.** To apply label propagation to our dumbbell complex example, we extend the label propagation algorithm of [75] to simplicial complexes, again, by replacing the vertex-vertex affinity matrix with edge-edge affinity matrix $A_d$. The resulting stochastic matrix $P = A_d \Delta_d^{-1}$ captures the transition probabilities between edges instead of vertices. Without considering the orientation of edges or triangles, the algorithm can be considered as applying label propagation to the dual graph of the simplicial complex.

In addition to the example shown in Figure 6, we give a few more instances of the results of label propagation on the dumbbell complex in Figure 8 with different initial labels.

## 6 Discussion

We present an algorithm for the simplification of simplicial complexes that preserves the spectral properties of the up Laplacian. Our work is strongly motivated by the study of an emerging class of learning algorithms based on simplicial complexes, and in particular, those spectral algorithms that operate with higher order Laplacians. We would like to understand the benefits and incurred error when such learning algorithms are applied to sketches of the data. Several ongoing and future directions are described below.

**Challenges for efficient implementation.** To compute generalized effective resistances



**(a)** **(b)**

**(c)** **(d)**

Figure 8: More instances of label propagation on the dumbbell complex before **(a), (c)** and after **(b), (d)** sparsification.

of $i$-simplices of $K$ exactly, we need to solve linear systems involving the up Laplacian. The best solution is to compute a QR or SVD decomposition of the scaled incidence matrix $W_i^{1/2} D_{i-1}$, which can be done in $O(n_i \cdot n_{i-1}^2)$ time, where $n_i$ is the number of $i$-simplices and $n_{i-1}$ is the number of $(i-1)$-simplices of $K$.

Spielman and Srivastava [63] gave an algorithm that can approximate effective resistances and produce the sparse graph in $O(m \log(r)/\varepsilon^2)$ time, where $m$ is the number of edges, and $r$ is the ratio of the largest to the smallest edge weights. The key to their algorithm was an efficient SDD solver [66] that approximately solves the linear systems involving the graph Laplacian in $\tilde{O}(m \log(1/\delta))$, where $\delta$ is an error parameter. Recent SDD solvers, also using graph-based preconditioners (low stretch spanning trees, etc.), have improved the running time even further [39, 18].

However, the up Laplacians $\mathcal{L}_{K,i}$ for $i \geq 1$ are not diagonally dominant matrices. Therefore, these fast SDD solvers may not be applied directly to approximate generalized effective resistance. Although solving linear systems of higher order Laplacians has been studied for limited classes of complexes [17, 19, 42], no such solvers exist for up Laplacians of arbitrary (nongeometric) simplicial complexes.

Generalizations of spanning trees to higher dimensions may be useful in constructing fast solvers for up Laplacians of arbitrary simplicial complexes. Alternatively, sparsification using generalized effective resistance can be thought of as a form of leverage score sampling. Computation of exact leverage scores has the same time complexity as the computation of exact generalized effective resistance. However, fast algorithms to compute constant factor approximations to leverage scores [26, 16] exist, which in theory, can run in $o(n_i \cdot n_{i-1}^2)$ time. However, further analysis is required before we can apply any of these approaches to approximate generalized effective resistances and make claims about the runtime of such implementations.

**Physical meaning of generalized effective resistance.** We believe the generalization of effective resistance to simplicial complexes, introduced in Section 3.1, may find other applications in analyzing simplicial complexes. Although the generalization is algebraically straightforward, its interpretation and properties pose many natural and interesting questions. For example, does it have an interpretation in terms of a random process, such as an effective commute time as in the case of a graph (see, e.g., [29])? Is it related to minimum spanning objects in the simplicial complex? Does it play a further role in spectral clustering of simplicial complexes?

**Multilevel and Hodge sparsification.** We are also interested in performing multilevel sparsification of simplicial complexes; for example, we would like to sparsify triangles and edges simultaneously while preserving spectral properties of the dimension-0 and dimension-1 up Laplacians. Such sparsification is challenging if we would like to simultaneously maintain structures of simplicial complexes; we may be able to relax our structural constraints to work with hypergraphs instead. In addition, multilevel sparsification is also related to preserving the spectral properties of the (Hodge) Laplacian. Finally, we are also interested in deriving formal connections between homological sparsification and spectral sparsification of simplicial complexes.

## References

[1] R. Andersen, F. Chung, and K. Lang. Local graph partitioning using pagerank vectors. *IEEE Symposium on Foundations of Computer Science*, 2006.

[2] R. Andersen and K. J. Lang. Communities from seed sets. *International Conference on the World Wide Web*, pages 223–232, 2006.

[3] K. L. Anderson, J. S. Anderson, S. Palande, and B. Wang. Topological data analysis of functional mri connectivity in time and space domains. *Proceedings International Workshop on Connectomics in NeuroImaging (CNI) at MICCAI*, 2018.

[4] J. Batson, D. A. Spielman, N. Srivastava, and S.-H. Teng. Spectral sparsification of graphs: theory and algorithms. *Communications of the ACM*, 56(8):87–94, 2013.

[5] A. A. Benczúr and D. R. Karger. Approximating s-t minimum cuts in $\tilde{O}(n^2)$ time. *ACM Symposium on Theory of Computing*, pages 47–55, 1996.

[6] P. Bendich, E. Gasparovic, J. Harer, R. Izmailov, and L. Ness. Multi-scale local shape analysis for feature selection in machine learning applications. *International Joint Conference on Neural Networks*, 2015.

[7] A. Benson, D. F. Gleich, and J. Leskovec. Tensor spectral clustering for partitioning higher-order network structures. *SIAM International Conference on Data Mining*, pages 118–126, 2015.

[8] M. B. Botnan and G. Spreemann. Approximating persistent homology in euclidean space through collapses. *Applicable Algebra in Engineering, Communication and Computing*, pages 1–29, 2015.

[9] M. Buchet, F. Chazal, S. Y. Oudot, and D. R. Sheehy. Efficient and robust persistent homology for measures. *ACM-SIAM Symposium on Discrete Algorithms*, pages 168–180, 2015.

[10] G. Carlsson. Topology and data. *Bulletin of the American Mathematical Society*, 46(2):255–308, 2009.

[11] B. Cassidy, C. Rae, and V. Solo. Brain activity: Conditional dissimilarity and persistent homology. *International Symposium on Biomedical Imaging*, pages 1356–1359, 2015.

[12] N. J. Cavanna, M. Jahanseir, and D. R. Sheehy. A geometric perspective on sparse filtrations. *Proceedings Canadian Conference on Computational Geometry*, 2015.

[13] A. K. Chandra, P. Raghavan, W. L. Ruzzo, R. Smolensky, and P. Tiwari. The electrical resistance of a graph captures its commute and cover times. *Computational Complexity*, 6:312–340, 1996.

[14] A. Choudhary, M. Kerber, and S. Raghvendra. Polynomial-sized topological approximations using the permutahedron. *arXiv: 1601.02732*, 2016.

[15] F. R. K. Chung. *Spectral Graph Theory*, volume 92. American Mathematical Society, 1997.

[16] K. L. Clarkson and D. P. Woodruff. Low Rank Approximation and Regression in Input Sparsity Time. *Journal of the ACM*, 63(6):1–45, 2012.

[17] M. B. Cohen, B. T. Fasy, G. L. Miller, A. Nayyeri, R. Peng, and N. Walkington. Solving 1-Laplacians in nearly linear time: collapsing and expanding a topological ball. *ACM-SIAM Symposium on Discrete Algorithms*, pages 204–216, 2014.

[18] M. B. Cohen, R. Kyng, G. L. Miller, J. W. Pachocki, R. Peng, A. B. Rao, and S. C. Xu. Solving sdd linear systems in nearly mlog1/2n time. *Proceedings ACM Symposium on Theory of Computing*, pages 343–352, 2014.

[19] S. I. Daitch and D. A. Spielman. Support-graph preconditioners for 2-dimensional trusses. *SIAM Workshop on Combinatorial Scientific Computing*, 2007.

[20] V. de Silva and R. Ghrist. Coverage in sensor networks via persistent homology. *Algebraic and Geometric Topology*, 7:339–358, 2007.

[21] T. K. Dey, F. Fan, and Y. Wang. Computing topological persistence for simplicial maps. *Symposium on Computational Geometry*, pages 345–354, 2014.

[22] T. K. Dey, F. Fan, and Y. Wang. Graph induced complex on point data. *Computational Geometry*, 48(8):575–588, 2015.

[23] R. Diestel. *Graph Theory*. Springer Graduate Texts in Mathematics, 2000.

[24] D. Dotterrer and M. Kahle. Coboundary expanders. *Journal of Topology and Analysis*, 4:499–514, 2012.

[25] P. G. Doyle and J. L. Snell. *Random walks and electric networks*. Mathematical Association of America, 1984.

[26] P. Drineas, M. Magdon-Ismail, M. W. Mahoney, and D. P. Woodruff. Fast approximation of matrix coherence and statistical leverage. *Journal of Machine Learning Research*, 13(1):3475–3506, 2012.

[27] H. Edelsbrunner and J. Harer. Persistent homology - a survey. *Contemporary Mathematics*, 453:257–282, 2008.

[28] H. Edelsbrunner, D. Letscher, and A. Zomorodian. Topological persistence and simplification. *Discrete & Computational Geometry*, 4(28):511–533, 2002.

[29] A. Ghosh, S. Boyd, and A. Saberi. Minimizing effective resistance of a graph. *SIAM Review*, 50(1):37–66, 2008.

[30] R. Ghrist. Barcodes: the persistent topology of data. *Bullentin of the American Mathematical Society*, 45(1):61–75, 2008.

[31] R. Ghrist. *Elementary Applied Topology*. Createspace, 2014.

[32] D. F. Gleich. Pagerank beyond the web. *SIAM Review*, 57(3):321–363, 2015.

[33] D. F. Gleich, L.-H. Lim, and Y. Yu. Multilinear pagerank. *SIAM Journal on Matrix Analysis and Applications*, 36(4):1507–1541, 2015.

[34] L. J. Grady and J. Polimeni. *Discrete Calculus: Applied Analysis on Graphs for Computational Science*. Springer, 2010.

[35] A. Gundert and M. Szedlák. Higher dimensional discrete Cheeger inequalities. *Journal of Computational Geometry*, 6(2):54–71, 2015.

[36] J. Haupt, X. Li, and D. P. Woodruff. Near optimal sketching of low-rank tensor regression. *Advances in Neural Information Processing Systems*, pages 3466–3476, 2017.

[37] D. Horak and J. Jost. Spectra of combinatorial Laplace operators on simplicial complexes. *Advances in Mathematics*, 244:303–336, 2013.

[38] X. Jiang, L.-H. Lim, Y. Yao, and Y. Ye. Statistical ranking and combinatorial hodge theory. *Mathematical Programming*, 127(1):203–244, 2011.

[39] J. A. Kelner, L. Orecchia, A. Sidford, and Z. A. Zhu. A simple, combinatorial algorithm for solving sdd systems in nearly-linear time. *ACM symposium on Theory of computing*, pages 911–920, 2013.

[40] M. Kerber and R. Sharathkumar. Approximate Cěch complexes in low and high dimensions. *Proceedings Symposium on Algorithms and Computation, LNCS*, 8283:666–676, 2013.

[41] T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, 2009.

[42] R. Kyng, R. Peng, R. Schwieterman, and P. Zhang. Incomplete nested dissection. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018*, pages 404–417, 2018.

[43] A. B. Lee, K. S. Pedersen, and D. Mumford. The nonlinear statistics of high-contrast patches in natural images. *International Journal of Computer Vision*, 54(1-3):83–103, 2003.

[44] H. Lee, M. K. Chung, H. Kang, B.-N. Kim, and D. S. Lee. Discriminative persistent homology of brain networks. *International Symposium on Biomedical Imaging*, pages 841–844, 2011.

[45] H. Lee, H. Kang, M. K. Chung, B.-N. Kim, and D. S. Lee. Persistent brain network homology from the perspective of dendrogram. *IEEE Transactions on Medical Imaging*, 31(12):2267–2277, 2012.

[46] A. Lubotzky. Ramanujan complexes and high dimensional expanders. *Japanese Journal of Mathematics*, 9:137–169, 2014.

[47] M. W. Mahoney, L. Orecchia, and N. K. Vishnoi. A local spectral method for graphs: With applications to improving graph partitions and exploring data graphs locally. *Journal of Machine Learning Research*, 13:2339–2365, 2012.

[48] S. Mukherjee and J. Steenbergen. Random walks on simplicial complexes and harmonics. *Random Structures & Algorithms*, 2016.

[49] J. R. Munkres. *Elements of algebraic topology.* Addison-Wesley, Redwood City, CA, USA, 1984.

[50] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. *Advances In Neural Information Processing Systems*, 2001.

[51] N. H. Nguyen, P. Drineas, and T. D. Tran. Tensor sparsification via a bound on the spectral norm of random tensors. *Information and Inference*, 4(3):195–229, 2015.

[52] M. Nicolau, A. J. Levine, and G. Carlsson. Topology based data analysis identifies a subgroup of breast cancers with a unique mutational profile and excellent survival. *Proceedings of the National Academy of Sciences*, 108(17):7265–7270, 2011.

[53] B. Osting, C. Brune, and S. J. Osher. Optimal data collection for informative rankings expose well-connected graphs. *Journal of Machine Learning Research*, 15:2981–3012, 2014.

[54] B. Osting, C. D. White, and E. Oudet. Minimal Dirichlet energy partitions for graphs. *SIAM Journal on Scientific Computing*, 36(4):A1635–A1651, 2014.

[55] B. Osting, Y. Yao, J. Xiong, and Q. Xu. Analysis of crowdsourced sampling strategies for hodgerank with sparse random graphs. *Applied and Computational Harmonic Analysis*, 41(2):540–560, 2016.

[56] O. Parzanchevski and R. Rosenthal. Simplicial complexes: Spectrum, homology and random walks. *Random Structures & Algorithms*, 2016.

[57] O. Parzanchevski, R. Rosenthal, and R. J. Tessler. Isoperimetric inequalities in simplicial complexes. *Combinatorica*, 36(2):195–227, 2016.

[58] J. A. Perea and J. Harer. Sliding windows and persistence: An application of topological methods to signal analysis. *Foundations of Computational Mathematics*, 15(3):799–838, 2015.

[59] W. Ren, Q. Zhao, R. Ramanathan, J. Gao, A. Swami, A. Bar-Noy, M. P. Johnson, and P. Basu. Broadcasting in multi-radio multi-channel wireless networks using simplicial complexes. *Wireless networks*, 19(6):1121–1133, 2013.

[60] M. Rudelson and R. Vershynin. Sampling from large matrices: An approach through geometric functional analysis. *Journal of the ACM*, 54(4):21, 2007.

[61] D. Sheehy. Linear-size approximations to the Vietoris-Rips filtration. *Discrete & Computational Geometry*, 49(4):778–796, 2013.

[62] Z. Song, D. P. Woodruff, and P. Zhong. Relative error tensor low rank approximation. *ACM-SIAM Symposium on Discrete Algorithms*, pages 2772–2789, April 2017.

[63] D. A. Spielman and N. Srivastava. Graph sparsification by effective resistances. *SIAM Journal on Computing*, 40(6):1913–1926, 2011.

[64] D. A. Spielman and S.-H. Teng. Spectral sparsification of graphs. *SIAM Journal on Computing*, 40(4):981–1025, 2011.

[65] D. A. Spielman and S.-H. Teng. A local clustering algorithm for massive graphs and its application to nearly-linear time graph partitioning. *SIAM Journal on Computing*, 42(1):1–26, 2013.

[66] D. A. Spielman and S.-H. Teng. Nearly linear time algorithms for preconditioning and solving symmetric, diagonally dominant linear systems. *SIAM Journal on Matrix Analysis and Applications*, 35(3):835–885, 2014.

[67] J. Steenbergen, C. Klivans, and S. Mukherjee. A Cheeger-type inequality on simplicial complexes. *Advances in Applied Mathematics*, 56:56–77, 2014.

[68] M. Szummer and T. Jaakkola. Partially labeled classification with markov random walks. *Advances in neural information processing systems*, 14:945–952, 2002.

[69] A. Tausz and G. Carlsson. Applications of zigzag persistence to topological data analysis. *arxiv:1108.3545*, 2011.

[70] Y. van Gennip, N. Guillen, B. Osting, and A. L. Bertozzi. Mean curvature, threshold dynamics, and phase field theory on finite graphs. *Milan Journal of Mathematics*, 82(1):3–65, 2014.

[71] U. von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.

[72] B. Wang, B. Summa, V. Pascucci, and M. Vejdemo-Johansson. Branching and circular features in high dimensional data. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):1902–1911, 2011.

[73] Y. Wang, H.-Y. Tung, A. Smola, and A. Anandkumar. Fast and guaranteed tensor decomposition via sketching. *Advances in neural information processing systems*, pages 991–999, 2015.

[74] E. Wong, S. Palande, B. Wang, B. Zielinski, J. Anderson, and P. T. Fletcher. Kernel partial least squares regression for relating functional brain network topology to clinical measures of behavior. *International Symposium on Biomedical Imaging*, 2016.

[75] X. Zhu and Z. Ghahramani. Learning from labeled and unlabeled data with label propagation. Technical Report Technical Report CMU-CALD-02-107, Carnegie Mellon University, 2002.

[76] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. *International Conference on Machine Learning*, pages 912–919, 2003.

## A Implementation and Execution Rates

We will first take a quick look at the execution rates for the spectral clustering and label propagation algorithms used in our experiments. It is not our goal to provide efficient algorithms for spectral clustering or label propagation. We only want to show that, by reducing $n_k$, the number of $k$-simplices, our sparsification algorithm can greatly improve the rate of execution of both spectral clustering and label propagation. Since the spectrum of $\mathcal{L}_{K,k-1}$ is also approximately preserved, the output clusters or labels are approximately the same for the original simplicial complex and the sparse complex output by the sparsifier.

### A.1 Spectral Clustering

In spectral clustering, our objective is to cluster $(k-1)$-simplices of $K$ into $d$ clusters. To do this, we first compute the eigenvectors corresponding to the $d$ largest eigenvalues of the

Laplacian of the dual graph and then apply $k$-means clustering to $n_{k-1}$ points in $\mathbb{R}^d$ formed by these $d$ eigenvectors. To compute eigenvectors of a sparse, symmetric Laplacian matrix, one can use ARPACK's Implicitly Restarted Arnoldi Method (IRAM). The rates of execution (in flops) for various steps in the algorithm are as follows:

1. Computing Lanczos vectors and tridiagonal matrix - $O(nnz \cdot n_{k-1})$.
2. Eigendecomposition of the tridiagonal matrix - $O(n_{k-1}^2)$.
3. Computing $d$ eigenvectors of the input matrix - $O(n_{k-1} \cdot d)$.
4. $k$-means clustering with the $d$ eigenvectors - $O(n_{k-1} \cdot d \cdot t)$.

Here $nnz$ is the number of nonzero entries in the input matrix, and $t$ is the number of iterations required for the $k$-means (Lloyd's) algorithm to converge. The input matrix is the $n_{k-1} \times n_{k-1}$ Laplacian of the dual graph. The term $O(nnz \cdot n_{k-1})$ dominates the overall cost since $d \ll n_{k-1} < nnz$. Our sparsification algorithm can help reduce this cost significantly by reducing $nnz$.

### A.2 Label Propagation

In the label propagation problem, we are given discrete labels for a small subset of $(k-1)$-simplices of $K$, and the objective is to learn the labels on the remaining unlabeled $(k-1)$-simplices. Our label propagation algorithm requires constructing the adjacency matrix of $(k-1)$-simplices and then normlizing it to obain the transition probability matrix $P$, which can be achieved in $O(n_k)$ flops. The algorithm then requires solving the linear system $(I - P_{uu})y_u = P_{ul}y_l$, where, $y_l$ is the vector of known labels, and $P_{uu}$ is the submatrix of $P$ corresponding to unlabeled $(k-1)$-simplices. As long as the simplices are label-connected, i.e., there is a sequence of $k$-simplices connecting every unlabeled $(k-1)$-simplex to a labeled $(k-1)$-simplex, $(I - P_{uu})$ is symmetric positive definite. Using conjugate gradients, the system can be solved in $O(nnz \cdot n_{k-1})$, where $nnz$ is the number of nonzero entries in $P$. Once again, our sparsification algorithm can reduce the cost significantly by reducing $nnz$, the number of nonzero entries in $P$, which is proportional to the number of $k$-simplices.

### A.3 Sparsification

To sparsify a weighted simplicial complex $K$ at dimension $k$, we need to compute the generalized effective resistances of the $k$-simplices, which we defined as the diagonal entries of matrix $R_k = D_{k-1}(\mathcal{L}_{K,k-1})^+ D_{k-1}^T$. $\mathcal{L}_{K,k-1}$ is an $n_{k-1} \times n_{k-1}$ symmetric positive semi-definite matrix. $D_{k-1}$ is an $n_k \times n_{k-1}$ matrix. Both these matrices can be constructed in $O(n_k)$, where $n_k$ is the number of $k$-simplices of $K$.

**Computing generalized effective resistances exactly.** In a naïve implementation, one would obtain generalized effective resistances by computing $R_k$ as defined. To do that, we would first need to solve $n_k$ linear systems of the form $\mathcal{L}_{K,k-1}x_f = d_f^T$, where $d_f$ is the row of $D_{k-1}$ corresponding to $k$-simplex $f$ of $K$. The total cost of obtaining generalized effective resistances this way is as follows:

1. Obtain LU or QR decomposition of $L_{K,k-1}$ - $O(n_{k-1}^3)$.

2. Solve $n_k$ linear systems using the decomposition - $O(n_k \cdot n_{k-1}^2)$.

3. Compute $R_k$ - $O(n_k^2)$.

4. Obtain $q$ samples with replacement from the probability distribution defined by generalized effective resistances - $O(q) = O(n_{k-1} \log(n_{k-1})/\varepsilon^2)$.

Note that $\varepsilon$ is a fixed error parameter and $n_{k-1} < n_k$ (otherwise, sparsifying at dimension $k$ would not be necessary). Therefore, the total cost is dominated by the $O(n_k \cdot n_{k-1}^2)$ term.

Another possible approach is to look at sparsification by generalized effective resistances as a form of leverage score sampling. For an $n \times d$ matrix $A$, with SVD $A = U\Sigma V^T$, the leverage scores of rows of $A$ are defined as the squared norms of rows of $U$. To see the relation between generalized effective resistances and leverage scores, define $\Phi = W_k^{1/2} D_{k-1}$ to be a scaled incidence matrix with singular value decomposition $\Phi = U\Sigma V^T$. Note that $\mathcal{L}_{K,k-1} = \Phi^T \Phi$. The sampling probability of an $k$-dimensional simplex $f$ of $K$ is proportional to $\Pi(f,f) = w(f)R_k(f,f)$, where $\Pi = W_k^{1/2} R_k W_k^{1/2}$ is the projection matrix defined in the proof of theorem 3.1. Then, we have

$$\begin{aligned}
\Pi &= W_k^{1/2} D_{k-1} (\mathcal{L}_{K,k-1})^+ D_{k-1}^T W_k^{1/2} \\
&= \Phi(\Phi^T\Phi)^+\Phi^T \\
&= U\Sigma V^T (V\Sigma U^T U\Sigma V^T)^+ V\Sigma U^T \\
&= U\Sigma V^T (V\Sigma^2 V^T)^+ V\Sigma U^T \\
&= U\Sigma V^T (V^T)^+ (\Sigma^2)^{pinv} (V)^+ V\Sigma U^T \\
&= UU^T.
\end{aligned}$$

Therefore, diagonal entries of $\Pi$ are the same as diagonal entries of $UU^T$, which are precisely the leverage scores of rows of $\Phi$. Unfortunately, this approach requires computing the singular value decomposition of the $n_k \times n_{k-1}$ matrix $\Phi$, which again has the cost $O(n_k \cdot n_{k-1}^2)$.

**Approximating generalized effective resistances.** Corollary 3.1 is a straightforward generalization of [Spielman and Srivastava[63], Corollary 6]. It shows that a constant factor approximation of generalized effective resistances is sufficient to obtain a good sparsifier of $K$. Using this corollary, Spielman and Srivastava [63] gave an algorithm for graph sparsification that runs in $O(m \log(r)/\varepsilon^2)$ time, where $m$ is the number of edges and $r$ is the ratio of largest to smallest edge weights. The key to their algorithm was an efficient SDD solver [66] that approximately solves the linear system involving the graph Laplacian in $O(m \log(1/\delta))$ where $\delta$ is an error parameter. More recent SDD solvers, using graph-based preconditioners (low stretch spanning trees, etc.), have improved the running time even further [39]. The fastest known SDD solver, proposed by Cohen et al. [18], has $O(m \cdot \log^{1/2} n \log(1/\varepsilon))$ time complexity for an $n \times n$ SDD matrix with $m$ nonzero entries.

However, $\mathcal{L}_{K,k-1}$ for $k > 1$ is *not* a diagonally dominant matrix. Therefore, the fast SDD solvers used to compute effective resistances in graph sparsification cannot be used directly to compute generalized effective resistance. Solving linear systems in the

$k$-dimensional up Laplacian has been studied by Cohen et al. [17] for limited classes of complexes. A related line of work on spectral algorithms for 2-dimensional truss matrices was initiated by Daitch and Spielman [19]. However, more analysis is required before we can apply these approaches to approximate generalized effective resistances for simplices in arbitrary (nongeometric) simplicial complexes.

Some work has also been done on fast approximation of leverage scores [26, 16]. Given an $n \times d$ matrix with $n > d$, constant factor approximations of its leverage scores can be computed using randomization techniques like matrix sketching. These algorithms run in $O(nnz \cdot \log(n) + d^3 \cdot \log(d) \cdot \log(n)$. In fact, under certain conditions on $n$ and $d$, these methods can approximate leverage scores in $o(n \cdot d^2)$. However, further analysis is required to determine whether our input boundary matrices satisfy the specific assumptions under which this runtime bound holds. If they do, this runtime would translate to $o(n_k \cdot n_{k-1}^2)$ in our case.