Spectral relaxations of persistent rank invariants

Matt Piekenbrock & Jose A. Perea

Abstract

Using the fact that the persistent rank invariant determines the persistence diagram and vice versa, we introduce a framework for constructing families of continuous relaxations of the persistent rank invariant for persistence modules indexed over the real line. Like the rank invariant, these families obey inclusion-exclusion, are derived from simplicial boundary operators, and encode all the information needed to construct a persistence diagram. Unlike the rank invariant, these spectrally-derived families enjoy a number of stability and continuity properties typically reserved for persistence diagrams, such as smoothness and differentiability. By leveraging its relationship with combinatorial Laplacian operators, we find the non-harmonic spectra of our proposed relaxation encode valuable geometric information about the underlying space, prompting several avenues for geometric data analysis. As these Laplacian operators are trace-class operators, we also find the corresponding relaxation can be efficiently approximated with a randomized algorithm based on the stochastic Lanczos quadrature method. We investigate the utility of our relaxation with applications in topological data analysis and machine learning, such as parameter optimization and shape classification.

1 Introduction

Persistent homology [17] (PH) is the most widely deployed tool for data analysis and learning applications within the topological data analysis (TDA) community. Persistence-related pipelines often follow a common pattern: given a data set X as input, construct a simplicial complex K and an order-preserving function $f : K \to \mathbb{R}$ such that useful topological/geometric information may be gleaned from its *persistence diagram*—a multiset summary of f formed by pairs $(a, b) \in \mathbb{R}^2$ exhibiting non-zero *multiplicity* $\mu_p^{a,b} \in \mathbb{Z}_+$:

$$\mathrm{dgm}_p(K, f) \triangleq \{\, (a, b) : \mu_p^{a,b} \neq 0 \,\}, \qquad \mu_p^{a,b} \triangleq \min_{\delta>0} \left(\beta_p^{a+\delta,b-\delta} - \beta_p^{a+\delta,b+\delta}\right) - \left(\beta_p^{a-\delta,b-\delta} - \beta_p^{a-\delta,b+\delta}\right) \tag{1.1}$$

where $\beta_p^{a,b}$ is the rank of the linear map in homology induced by the inclusion $f^{-1}(-\infty, a] \hookrightarrow f^{-1}(-\infty, b]$. The surprising and essential quality of persistence is that these pairings exist, are unique, and are stable under additive perturbations [12]. Whether for shape recognition [8], dimensionality reduction [32], or time series analysis [28], persistence is the de facto connection between homology and the application frontier.

Though theoretically sound, diagrams suffer from many practical issues: they are sensitive to outliers, far from injective, and expensive both to compute *and* compare. Towards ameliorating these issues, practitioners have equipped diagrams with additional structure by way of maps to function spaces; examples include persistence images [1], persistence landscapes [6], and template functions [29]. Tackling the issue of injectivity, Turner et al. [34] propose an injective shape statistic of directional diagrams associated to a data set $X \subset \mathbb{R}^d$, sparking both an inverse theory for persistence and a mathematical foundation for metric learning. Despite the potential these extensions have in learning applications, diagrams can still be expensive to obtain. The severity of this issue is compounded in the parameterized setting, where adaptations of the persistence computation has proven non-trivial [30].

We seek to shift the computational paradigm on persistence while retaining its application potential: rather than following a construct-then-vectorize approach, we devise a spectral method that performs both steps, simultaneously and approximately. Our strategy is motivated both by a technical observation that suggests advantages exist for the rank invariant computation (section 2.1) and by measure-theoretic results on $\mathbb{R}$-indexed persistence modules [7, 9], which generalize (1.1) to rectangles $R = [a, b] \times [c, d]$ in the plane:

$$\mu_p^R(K, f) \triangleq \mathrm{card}\left(\mathrm{dgm}_p(K, f)\big|_R\right) = \beta_p^{b,c} - \beta_p^{a,c} - \beta_p^{b,d} + \beta_p^{a,d} \tag{1.2}$$

Notably, our approach not only avoids explicitly constructing diagrams, but is in fact *matrix-free*, circumventing the reduction algorithm from [16] entirely. Additionally, the relaxation is computable in linear space and quadratic time, can be iteratively approximated, and requires no complicated data structures or maintenance procedures to implement.

Contributions: Our primary contribution is the introduction of several families of spectral approximations to the rank invariants—μ_p and β_p**??**—all of which are Lipshitz continuous**??**, stable under relative perturbations**??**, and differentiable on the positive semi-definite cone**??**. By a reduction to spectral methods for Laplacian operators**??**, we also show these approximations are computable in $\approx O(m)$ memory and $\approx O(mn)$ time, where n, m are the number of $p, p+1$ simplices in K, respectively (see section 4). Moreover, both relaxations admit iterative $(1-\epsilon)$-approximation schemes**??**, and in both cases are recovered exactly when the parameters ϵ and τ made small enough**??**.

Outline: We now outline the proposed relaxation, leaving the rest of the paper to discuss theoretical and practical details. Informally, we study a family of vector-valued mappings over a *parameter space* $\mathcal{A} \subset \mathbb{R}^d$:

$$(X_\alpha, \mathcal{R}, \tau, \epsilon) \mapsto \mathbb{R}^h \tag{1.3}$$

where X_α is an $\mathcal{A}$-parameterized input data set, $\mathcal{R} \subset \Delta_+ = \{\, (a,b) \in \mathbb{R}^2 : a \leq b \,\}$ is a region which decomposes as a disjoint union of rectangles $R_1 \cup \ldots \cup R_h$—we will call such a set a *sieve*—and $(\tau, \epsilon) \in \mathbb{R}_+^2$ are smoothness/approximation parameters, respectively. The intuition is that $\mathcal{R}$ is used to filter and summarize the topological and geometric behavior exhibited by X_α for all $\alpha \in \mathcal{A}$, thereby *sifting* the diagrams in the space $\mathcal{A} \times \Delta_+$. The steps to produce this mapping are as follows:

1. Let K denote a fixed simplicial complex constructed from the data set X. Select a parameter space $\mathcal{A} \subset \mathbb{R}^d$ which indexes a family of filter functions $\{\, f_\alpha : K \to \mathbb{R} : \alpha \in \mathcal{A} \,\}$ of K, where:

$$f_\alpha(\tau) \leq f_\alpha(\sigma) \;\forall\, \tau \subseteq \sigma \in K \quad \text{and} \quad f_\alpha(\sigma) \text{ is continuous in } \alpha \in \mathcal{A} \text{ for every } \sigma \in K \tag{1.4}$$

   Exemplary choices of f_α include filtrations geometrically realized from methods that themselves are have parameters, such as density filtrations or time-varying filtrations over dynamic metric spaces [21].

2. Select a *sieve* $\mathcal{R} = R_1 \cup \ldots \cup R_h \subset \Delta_+$. This choice is application-dependent and typically requires a priori knowledge, though in section 5 we give evidence that, when $\mathcal{R}$ is unknown, random sampling may be sufficient for vectorization or data exploration purposes.

3. Fix a homology dimension $p \geq 0$ and parameters $(\tau, \epsilon) \in \mathbb{R}_+^2$ representing how *smoothly* and *accurately* the relaxation $\hat{\mu}_p^{\mathcal{R}}$ (defined in step 5 below) should model the quantity:

$$\mu_p^{\mathcal{R}}(K, f_\alpha) \triangleq \mathrm{card}\left(\mathrm{dgm}_p(f_\alpha)\big|_{\mathcal{R}}\right) \tag{1.5}$$

4. Choose an $\mathcal{A}$-parameterized Laplacian operator $\mathcal{L}_p^{a,b} : \mathcal{A} \times C^p(K_b, K_a; \mathbb{R}) \to C^p(K_b, K_a; \mathbb{R})$ on the relative p-cochains of (K_b, K_b) and a τ-parameterized continuous function $\phi(\cdot, \tau) : \mathbb{R} \to \mathbb{R}_+$ which converges to sgn_+ as $\tau \to 0$. The choice of $\mathcal{L}$ (e.g. Kirchoff, random walk) determines the kind of geometric/topological information to extract from (K, f_α), while ϕ determines how that information is encoded.

5. Denote by $\Lambda(\mathcal{L}_{a,b}(\alpha))$ the spectrum of $\mathcal{L}_{a,b}$ at any $\alpha \in \mathcal{A}$ ordered in non-increasing order. Our relaxation approximates (1.5) by $(1 \pm \epsilon)$-approximating μ_p at each corner point (a,b) in the boundary of $\mathcal{R}$:

$$\mu_p^{\mathcal{R}}(\alpha) \overset{\epsilon}{\approx} \hat{\mu}_p^{\mathcal{R}}(\alpha) \triangleq \sum_{(a,b)} s_{a,b} \cdot \lVert \Phi_\tau\big(\mathcal{L}_p^{a,b}(\alpha)\big) \rVert_*$$

   where $\Phi_\tau(\cdot)$ is a τ-parameterized matrix function induced by ϕ (see section 3). In section **??**, we show that letting both $\tau \to 0$ and $\epsilon \to 0$ yields the multiplicity function $\mu_p^{\mathcal{R}}$ exactly.

The remaining steps of the relaxation depend on the application in mind. Applications looking to vectorize persistence information over random and highly structured complexes may benefit from the concentration of mass phenomenon known to occur their spectra; examples include topology-guided image denoising [31], shape classification under metric invariants [8], bifurcation detection in dynamical systems [29], and so on. The differentiability of our relaxation also suggests it may be used in topological optimization applications, i.e. optimization problems whose loss functions incorporate persistence information [26].

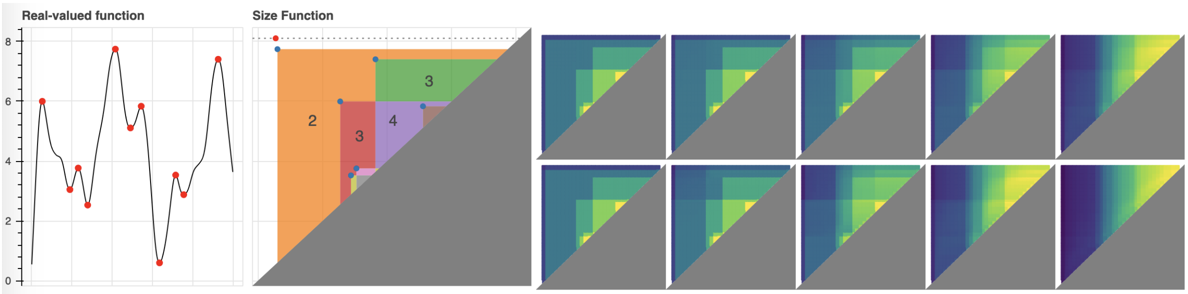

Figure 1: (left) A function $f : \mathbb{R} \to \mathbb{R}$ and its corresponding persistence diagram obtained by filtering f by its sublevel sets. (right) two spectral interpolations

2 Notation & Background

A *simplicial complex* $K \subseteq \mathcal{P}(V)$ over a finite set $V = \{v_1, v_2, \dots, v_n\}$ is a collection of simplices $\{\, \sigma : \sigma \in \mathcal{P}(V)\,\}$ such that $\tau \subseteq \sigma \in K \Rightarrow \tau \in K$. A *p-simplex* $\sigma \subseteq V$ is a set of $p+1$ vertices, the collection of which is denoted as K^p. An *oriented p-simplex* $[\sigma]$ is an ordered set $[\sigma] = (-1)^{|\pi|}\left[v_{\pi(1)}, v_{\pi(2)}, \dots, v_{\pi(p+1)}\right]$, where π is a permutation on $[\,p+1\,] = \{\,1, 2, \dots, p+1\,\}$ and $|\pi|$ the number of its inversions. The *p-boundary* $\partial_p[\sigma]$ of an oriented p-simplex $[\sigma] \in K$ is defined as the alternating sum of its oriented co-dimension 1 faces, which collectively for all $\sigma \in K^p$ define the p-th *boundary matrix* ∂_p of K:

$$\partial_p[i,j] \triangleq \begin{cases} (-1)^{s_{ij}} & \sigma_i \in \partial[\sigma_j] \\ 0 & \text{otherwise} \end{cases}, \qquad \partial_p[\sigma] \triangleq \sum_{i=1}^{p+1} (-1)^{i-1}[v_1, \dots, v_{i-1}, v_{i+1}, \dots v_{p+1}] \tag{2.1}$$

where $s_{ij} = \operatorname{sgn}([\sigma_i], \partial[\sigma_j])$ records the orientation. Extending (2.1) to all simplices in $\sigma \in K$ for all $p \leq \dim(K)$ yields the *full boundary matrix* ∂. With a small abuse in notation, we use ∂_p to denote both the boundary operator and its ordered matrix representative. When it is not clear from the context, we will clarify which representation is intended.

Generalizing beyond simplices, given a field $\mathbb{F}$, an *oriented p-chain* is a formal $\mathbb{F}$-linear combination of oriented p-simplices of K whose boundary $\partial_p[c]$ is defined linearly in terms of its constitutive simplices. The collection of p-chains under addition yields an $\mathbb{F}$-vector space $C_p(K)$ whose boundaries $c \in \partial_p[c']$ satisfying $\partial_p[c] = 0$ are called *cycles*. Together, the collection of p-boundaries and p-cycles forms the groups $B_p(K) = \operatorname{Im} \partial_{p+1}$ and $Z_p(K) = \operatorname{Ker} \partial_p$, respectively. The quotient space $H_p(K) = Z_p(K)/B_p(K)$ is called the *p-th homology group of* K with coefficients in $\mathbb{F}$ and its dimension β_p is the *p-th Betti number* of K.

A *filtration* is a pair (K, f) where $f : K \to I$ is a *filter function* over an index set I satisfying $f(\tau) \leq f(\sigma)$ whenever $\tau \subseteq \sigma$, for any $\tau, \sigma \in K$. For every pair $(a, b) \in I \times I$ satisfying $a \leq b$, the sequence of inclusions $K_a \subseteq \cdots \subseteq K_b$ induce linear transformations $h_p^{a,b} : H_p(K_a) \to H_p(K_b)$ at the level of homology. When $\mathbb{F}$ is a field, this sequence of homology groups uniquely decompose (K, f) into a pairing (σ_a, σ_b) demarcating the evolution of homology classes [36]: σ_a marks the creation of a homology class, σ_b marks its destruction, and the difference $|a - b|$ records the lifetime of the class, called its *persistence*. The persistent homology groups are the images of these maps and the persistent Betti numbers are their dimensions:

$$H_p^{a,b} = \begin{cases} H_p(K_a) & a = b \\ \operatorname{Im} h_p^{a,b} & a < b \end{cases}, \qquad \beta_p^{a,b} = \begin{cases} \beta_p(K_a) & a = b \\ \dim(H_p^{a,b}) & a < b \end{cases} \tag{2.2}$$

For a fixed $p \geq 0$, the collection of persistent pairs (a, b) together with unpaired simplices (c, ∞) form a summary representation $\mathrm{dgm}_p(K, f)$ called the *p-th persistence diagram of* (K, f). Conceptually, $\beta_p^{a,b}$ counts the number of persistent pairs lying inside the box $(-\infty, a\,] \times (\,b, \infty)$—the number of persistent homology groups born at or before a that died sometime after b. When a given quantity depends on fixed parameters that are irrelevant or unknown, we use an asterisk. Thus, $H_p^*(K)$ refers to any homology group of K.

We will at times need to generalize the notation given thus far to the *parameterized* setting. Towards this end, for some $\mathcal{A} \subseteq \mathbb{R}^d$, we define an *$\mathcal{A}$-parameterized filtration* as a pair (K, f_α) where K is a simplicial complex and $f : K \times \mathcal{A} \to \mathbb{R}$ an $\mathcal{A}$-parameterized filter function satisfying:

$$f_\alpha(\tau) \leq f_\alpha(\sigma) \;\forall\, \tau \subseteq \sigma \in K \quad \text{and} \quad f_\alpha(\sigma) \text{ is continuous in } \alpha \in \mathcal{A} \text{ for every } \sigma \in K \tag{2.3}$$

Intuitively, when $\mathcal{A} = \mathbb{R}$, one can think of α as a *time* parameter (see Figure 1) and each $f_\alpha(\sigma)$ as tracing a curve in $\mathbb{R}^2$ parameterized by α. Examples of parameterized filtrations include:

- (Constant filtration) For a filter $f : K \to \mathbb{R}$, let (K, f_α) denote the parameterized filtration obtained by declaring $f_\alpha(\sigma) = f(\sigma)$ for all $\alpha \in \mathcal{A}$ and all $\sigma \in K$. We refer to (K, f_α) as the *constant filtration.*
- (Dynamic Metric Spaces) For a finite set X, let $\gamma_X = (X, d_X(\cdot))$ denote a dynamic metric space [21], where $d_X(\cdot) : \mathbb{R} \times X \times X$ denotes a time-varying metric. For any fixed $K \subset \mathcal{P}(X)$, the pair (K, f_α) obtained by setting $f_\alpha(\sigma) = \max_{x,x' \in \sigma} d_X(\alpha)(x, x')$ recovers the notion of a *time-varying Rips filtration.*
- (Filter combinations) For $f, g : K \to \mathbb{R}$ filters over K, a natural family of filtrations (K, h_α) is obtained by $h_\alpha = (1 - \alpha)f + \alpha g$ for all $\alpha \in [0, 1]$, i.e. *convex combinations* of f and g.
- (Multi-filtrations) More generally, a d-dimensional filtration on the category of simplicial complexes **Simp** is a functor $\mathcal{F} : \mathbb{R}^d \to \textbf{Simp}$ satisfying $\mathcal{F}(a, b) : \mathcal{F}_a \to \mathcal{F}_b$ an inclusion for all $a \leq b$ []. By fixing $K \in \textbf{Simp}$ and letting $\mathcal{A} = \mathbb{R}^d$, we recover the notion of a *multi-filtration.*

2.1 Technical background

The following results summarize some technical observations motivating this effort, which will be used in several proofs. Though these observations are background material, they contextualize our non-traditional computation of the rank invariant (Corollary 2) and serve as the motivation for this work.

Among the most widely known results for persistence is the structure theorem [36], which shows 1-parameter persistence modules decompose in an *essentially unique* way. Computationally, the corresponding Pairing Uniqueness Lemma [13] asserts that if $R = \partial V$ decomposes the boundary matrix $\partial \in \mathbb{F}^{N \times N}$ to a *reduced* matrix $R \in \mathbb{F}^{N \times N}$ using left-to-right column operations, then:

$$R[i, j] \neq 0 \;\;\Leftrightarrow\;\; \mathrm{rank}(\partial^{i,j}) - \mathrm{rank}(\partial^{i+1,j}) + \mathrm{rank}(\partial^{i+1,j\text{-}1}) - \mathrm{rank}(\partial^{i,j\text{-}1}) \neq 0 \tag{2.4}$$

where $\partial^{i,j}$ denotes the lower-left submatrix defined by the first j columns and the last $m - i + 1$ rows (rows i through m, inclusive). Thus, the existence of non-zero "pivot" entries in R may be inferred entirely from the ranks of certain submatrices of ∂. Part of the validity of (2.4) can be attributed to the following Lemma:

Lemma 1. *Given filtration (K, f) of size $N = |K|$, let $R = \partial V$ denote the decomposition of the filtered boundary matrix $\partial \in \mathbb{F}^{N \times N}$. Then, for any pair (i, j) satisfying $1 \leq i < j \leq N$, we have:*

$$\mathrm{rank}(R^{i,j}) = \mathrm{rank}(\partial^{i,j}) \tag{2.5}$$

Equivalently, all lower-left submatrices of ∂ have the same rank as their corresponding submatrices in R.

An explicit proof of both of these facts can be found in [14], though the latter was also noted in passing by Edelsbrunner [17]. Though typically viewed as minor facts needed to prove the correctness of the reduction algorithm, the implications of these two observations are are quite general, as recently noted by [3]:

Corollary 1 (Bauer et al. [3])**.** *Any persistence algorithm which preserves the ranks of the submatrices $\partial^{i,j}(K, f)$ for all $i, j \in [N]$ is a valid persistence algorithm.*

Indeed, though R is not unique, its non-zero pivots are, and these pivots *define* the persistence diagram. Moreover, due to (2.5), both β_p^* and μ_p^* may be written as a sum of ranks of submatrices of ∂_p and ∂_{p+1}:

Corollary 2 ([10, 14])**.** *Given a fixed $p \geq 0$, a filtration (K, f) with filtration values $\{\, a_i \,\}_{i=1}^N$, and a rectangle $R = [a_i, a_j] \times [a_k, a_l] \subset \Delta_+$, the persistent Betti and multiplicity functions may be written as:*

$$\beta_p^{a_i,a_j}(K, f) = \mathrm{rank}(C_p(K_i)) - \mathrm{rank}(\partial_p^{1,i}) - \mathrm{rank}(\partial_{p+1}^{1,j}) + \mathrm{rank}(\partial_{p+1}^{i+1,j}) \tag{2.6}$$

$$\mu_p^R(K, f) = \mathrm{rank}(\partial_{p+1}^{j+1,k}) - \mathrm{rank}(\partial_{p+1}^{i+1,k}) - \mathrm{rank}(\partial_{p+1}^{j+1,l}) + \mathrm{rank}(\partial_{p+1}^{i+1,l}) \tag{2.7}$$

Though (2.7) was pointed out by Cohen-Steiner et al. in [13] and exploited computationally by Chen & Kerber in [10], to the authors knowledge the only explicit derivation and proof of (2.6) is given by Dey & Wang [14] (see section 3.3.1). For completeness, we give our own detailed proof of corollary 2 in the appendix. In practice, neither expressions seem used or even implemented in any commonly used persistence software.

There are two very important properties of the expressions from Corollary 2: (1) they are comprised strictly of *rank* computations, and (2) all terms involve *unfactored* boundary matrices. Coupled with measure-theoretic perspectives on persistence [9], the former property suggests variational perspectives of the rank function might yield interesting spectral relaxations of (2.7) and (2.6) useful for e.g. optimization purposes. Moreover, the latter property combined with Corollary 1 suggests alternative computation strategies may be employed to extract persistence information *without* using reduction. Indeed, combining these observations suggest several connections to other areas of applied mathematics, such as the rich theory of matrix functions [], the tools developed as part of "The Laplacian Paradigm" [], or the very recent connections between rank estimation and trace estimation []. The rest of the paper is dedicated to exploring these connections and their implications.

3 Spectral relaxation and its implications

In this section, we will introduce our proposed relaxation by successively relaxing and generalizing different aspects of (2.6). To motivate these relaxations, it is instructive to examine the how traditional expressions of the persistent rank invariants compare to those from Corollary 2. Given a filtration (K, f) of size $N = |K|$ with $f : K \to I$ defined over some index set I, its p-th persistent Betti number $\beta_p^{a,b}$ at index $(a, b) \in I \times I$, is defined as follows:

$$\begin{aligned}\beta_p^{a,b} &= \dim\left(Z_p(K_a)/B_p(K_b)\right)\\ &= \dim\left(Z_p(K_a)/(Z_p(K_a)\cap B_p(K_b))\right.\\ &= \dim\left(Z_p(K_a)\right) - \dim\left(Z_p(K_a)\cap B_p(K_b)\right)\end{aligned} \tag{3.1}$$

Computationally, observe that (3.1) reduces to one nullity computation and one subspace intersection computation. While the former is easy to re-cast as a spectral computation, computing the latter typically requires obtaining bases via matrix decomposition. Constructing these bases explicitly using conventional [4, 19] or persistence-based [24, 36] algorithms effectively[1] requires $\Omega(N^3)$ time and $\Omega(N^2)$ space. As the persistence algorithm also exhibits $O(N^3)$ time complexity and completely characterizes $\beta_p^{a,b}$ over *all* values $(a, b) \in I \times I$, there is little incentive to compute $\beta_p^{a,b}$ with such direct methods (and indeed, they are largely unused). Because of this, we will focus on expressions (2.6) and (2.7) throughout the rest of the paper.

3.1 Parameterized boundary operators

In typical dynamic persistence settings (e.g. [13]), a decomposition $R = \partial V$ of the boundary matrix ∂ must be permuted and modified frequently to maintain a total order with respect to f_α. In contrast, the rank function is permutation invariant, i.e. for any $X \in \mathbb{R}^{n\times n}$ and permutation P we have:

$$\mathrm{rank}(X) = \mathrm{rank}(P^T X P)$$

This suggests rank computations on boundary matrices need not maintain this ordering—as long as they have the same non-zero pattern as their filtered counterparts, their ranks will be identical. In this section, we exploit this fact by showing how the expressions from (2.6) and (2.7) may be made *permutation invariant.*

Let (K, f_α) denote parameterized family of filtrations of a simplicial complex of size $|K^p| = n$. Fix an arbitrary linear extension $(K, \preceq)$ of the face poset of K. Define the *$\mathcal{A}$-parameterized boundary operator* $\hat{\partial}_p(\alpha) \in \mathbb{R}^{n\times n}$ of (K, f_α) as the $n \times n$ matrix ordered by $\preceq$ for all $\alpha \in \mathcal{A}$ whose entries (k, l) satisfy:

$$\partial_p(\alpha)[k,l] = \begin{cases} s_{kl}\cdot f_\alpha(\sigma_k)\cdot f_\alpha(\sigma_l) & \text{if } \sigma_k \in \partial_p(\sigma_l)\\ 0 & \text{otherwise}\end{cases} \tag{3.2}$$

[1] Note about matrix multiplication constant

where $s_{kl} = \mathrm{sgn}([\sigma_k], \partial[\sigma_l])$ is the sign of the oriented face $[\sigma_k]$ in $\partial[\sigma_l]$. Observe that (1) the non-zero entries from (3.2) vary continuously in f_α and (2) $\partial_p(\alpha)$ decouples into a product of diagonal matrices $D_*(f_\alpha)$:

$$\partial_p(\alpha) \triangleq D_p(f_\alpha) \cdot \partial_p(K_\preceq) \cdot D_{p+1}(f_\alpha) \tag{3.3}$$

where $D_p(f_\alpha)$ and $D_{p+1}(f_\alpha)$ are diagonal matrices whose non-zero entries are ordered by restrictions of f_α to $K^p_\preceq$ and $K^{p+1}_\preceq$, respectively. Clearly, $\mathrm{rank}(\partial_p(\alpha)) = \mathrm{rank}(\partial_p(K_\preceq))$ when the diagonal entries of D_p and D_{p+1} are strictly positive. Moreover, observe we may restrict to those "lower left" matrices from Lemma 1 via post-composing step functions $\bar{S}_a(x) = \mathbb{1}_{x>a}(x)$ and $S_b(x) = \mathbb{1}_{x\leq b}(x)$ to D_p and D_{p+1}, respectively:

$$\hat{\partial}_p^{a,b}(\alpha) \triangleq D_p(\bar{S}_a \circ f_\alpha) \cdot \partial_p(K_\preceq) \cdot D_{p+1}(S_b \circ f_\alpha) \tag{3.4}$$

Though these step functions are discontinuous at their chosen thresholds a and b, we may retain the element-wise continuity of (3.3) by exchanging them with clamped *smoothstep* functions $\mathcal{S} : \mathbb{R} \to [0, 1]$ that interpolate the discontinuous step portion of S along a fixed interval $(a, a + \omega)$, for some $\omega > 0$ (see Figure 2).

These observations motivate our first relaxation. Without loss in generality, assume the orientation of the simplices induced by $(K, \preceq)$ is inherited from the order on the vertex set V. To simplify the notation, we write $A^x = A^{*,x}$ to denote the submatrix including all rows of A and all columns of A up to x.

Proposition 1. *Given (K, f_α), any rectangle $R = [a, b] \times [c, d] \subset \Delta_+$, and $\delta > 0$ the number satisfying $a + \delta < b - \delta$ from* (1.2) *the $\mathcal{A}$-parameterized invariants $\beta_p^{a,b} : \mathcal{A} \times K \to \mathbb{N}$ and $\mu_p^R : \mathcal{A} \times K \to \mathbb{N}$ defined by:*

$$\beta_p^{a,b}(\alpha) \triangleq \mathrm{rank}\big(D_p(S_a \circ f_\alpha)\big) - \mathrm{rank}\big(\,\hat{\partial}_p^{a}(\alpha)\,\big) - \mathrm{rank}\big(\,\hat{\partial}_{p+1}^{b}(\alpha)\,\big) + \mathrm{rank}\big(\,\hat{\partial}_{p+1}^{a+\delta,b}(\alpha)\,\big) \tag{3.5}$$

$$\mu_p^R(\alpha) \triangleq \mathrm{rank}\big(\,\hat{\partial}_{p+1}^{b+\delta,c}(\alpha)\,\big) - \mathrm{rank}\big(\,\hat{\partial}_{p+1}^{a+\delta,c}(\alpha)\,\big) - \mathrm{rank}\big(\,\hat{\partial}_{p+1}^{b+\delta,d}(\alpha)\,\big) + \mathrm{rank}\big(\,\hat{\partial}_{p+1}^{a+\delta,d}(\alpha)\,\big) \tag{3.6}$$

yield the correct quantities $\mu_p^R(K, f_\alpha) = \mathrm{card}\left(\mathrm{dgm}_p(f_\alpha)\big|_R\right)$ and $\beta_p^{a,b} = \dim(H_p^{a,b}(K, f_\alpha))$ for all $\alpha \in \mathcal{A}$.

For completeness, a proof of Proposition 1 is given in the appendix. Note that in (3.4), we write $\partial_p(K_\preceq)$ (as opposed to $\partial_p(K, f)$) to emphasize $\partial_p(K_\preceq)$ is ordered according to a fixed linear ordering $(K, \preceq)$. The distinction is necessary as evaluating the boundary terms from corollary 2 would require ∂ to be explicitly filtered in the total ordering induced by f_α—which varies in $\mathcal{A}$—whereas the expressions obtained by replacing the constitutive terms in (2.6) and (2.7) with (3.5) and (3.6), respectively, require no such explicit filtering.

3.2 Parameterized Laplacians

For the sake of generality, it is important to make the class of expressions for β_p^* and μ_p^* as large as possible. Towards this, we exploit another identity of the rank function applicable to zero-characteristic fields $\mathbb{F}$:

$$\mathrm{rank}(X) = \mathrm{rank}(XX^T) = \mathrm{rank}(X^TX), \quad \text{for all } X \in \mathbb{F}^{n\times m}$$

In the context of boundary operators, note that $\partial_1\partial_1^T$ is the well known *graph Laplacian* [11], indicating we may express $\beta_0^*(\alpha)$ and $\mu_0^*(\alpha)$ using the ranks of Laplacian[2] matrices. Indeed, as the eigenvalues of XX^T (or X^TX) are given by the squares of the singular values of X, we may study the singular values of boundary operators through the spectra of Laplacians.

For any simplicial complex K, there are three natural Laplacian operators: the *up*-Laplacian $L_p^{\mathrm{up}}(K)$, the *down*-Laplacian $L_p^{\mathrm{dn}}(K)$, and their sum, which we refer to as the *combinatorial* Laplacian $\Delta_p(K)$:

$$\Delta_p = \underbrace{\partial_{p+1} \circ \partial_{p+1}^T}_{L_p^{\mathrm{up}}} + \underbrace{\partial_p^T \circ \partial_p}_{L_p^{\mathrm{dn}}} \tag{3.7}$$

All three operators Δ_p, L_p^{up}, and L_p^{dn} are symmetric, positive semi-definite, and compact [24] and thus have real, non-negative eigenvalues—moreover, the multisets $\Lambda(L_p^{\mathrm{up}})$ and $\Lambda(L_{p+1}^{\mathrm{dn}})$ differ only in the multiplicities of zero, implying they must have identical ranks (see Theorem 2.2 and 3.1 of [20]). Thus, for rank computations, it suffices to consider only one of them.

[2] By convention, we define $\partial_p = 0$ for all $p \leq 0$.

Let (K, f_α) denote a parameterized family of filtrations of a simplicial complex K equipped with a fixed but arbitrary linear extension $\preceq$ of its face poset and fixed orientations $s(\sigma)$ inherited from the total order on the vertex set $(V, \preceq)$. Without loss of generality, we define the weighted p up-Laplacian $\mathcal{L}_p \triangleq L_p^{\mathrm{up}}$ at index (a, b) as follows:

$$\mathcal{L}_p^{a,b}(\alpha) \triangleq D_p(\bar{S}_a \circ f_\alpha) \cdot \partial_{p+1}(K_\preceq) \cdot D_{p+1}(S_b \circ f_\alpha) \cdot \partial_{p+1}^T(K_\preceq) \cdot D_p(\bar{S}_a \circ f_\alpha) \tag{3.8}$$

where $D_p(f)$ denotes a diagonal matrix whose entries represent the application of f to the p-simplices of K. As in (3.4), fixing step function S_a and $\bar{S}_b$ at values $a, b \in \mathbb{R}$ yields operators whose ranks correspond to the ranks of certain "lower-left" submatrices of the corresponding full boundary matrix ∂ of (K, f). In particular, if $R = \partial V$ is the decomposition of (K, f_α) for some fixed choice of $\alpha \in \mathcal{A}$, then for any pair $a, b \in \Delta_+$ there exists indices $i = \sum_{\sigma \in K}(S_a \circ f_\alpha)(\sigma)$ and $j = \sum_{\sigma \in K}(S_b \circ f_\alpha)(\sigma)$ such that:

$$\mathrm{rank}(R_{p+1}^{i,j}) = \mathrm{rank}(\partial_{p+1}^{i,j}) = \mathrm{rank}(\hat{\partial}_{p+1}^{a,b}) = \mathrm{rank}\left((\hat{\partial}_{p+1}^{a,b})(\hat{\partial}_{p+1}^{a,b})^T\right) = \mathrm{rank}(\mathcal{L}_p^{a,b}) \tag{3.9}$$

where the second last equality uses the identity $\mathrm{rank}(X) = \mathrm{rank}(X^T X)$. This confirms that we may substitute any of the parameterized boundary operators used in Proposition 1 with weighted Laplacian operators $\partial_{p+1}^* \mapsto \mathcal{L}_p^*$ equipped with the appropriate down- and up-step functions S_* and $\bar{S}_*$, respectively.

Remark 1. One may interpret sending a subset $S \subseteq K$ of p-simplices to 0 as restricting to a sub-complex $L = K \setminus S$, which suggests we may use the inclusion $L \hookrightarrow K$ to re-define e.g. (3.1) and its subsequent extensions (3.8) using *simplicial pairs* (L, K), as in [24]. However, our parameterized definition of a boundary operator (3.2) may not always correspond to a subcomplex in the sense that if two distinct p-simplices $\tau, \tau' \in K^p, \tau \neq \tau'$ sharing a coface $\sigma \in K^{p+1}$, we allow the situation where $f(\tau) > 0$ and $f(\sigma) > 0$ but $f(\tau') = 0$ (due to the arbitrary choice of $a, b \in \mathbb{R}$).

3.3 Spectral rank relaxation

Under mild assumptions on f_α, the entries of the boundary operators from (3.4) are continuous functions of α when S is substituted appropriately with a smoothstep relaxation. In contrast, due to the rank function, the quantities from Proposition 1 are integer-valued and thus discontinuous functions. To understand how these discontinuities arise, we consider the spectral characterization of the rank function:

$$\mathrm{rank}(X) = \sum_{i=1}^{n} \mathrm{sgn}_+(\sigma_i(X)), \qquad \mathrm{sgn}_+(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases} \tag{3.10}$$

In the above, $\{\sigma_i\}_{i=1}^n$ are the singular values $\Sigma = \mathrm{diag}(\{\sigma_i\}_{i=1}^n)$ from the singular value decomposition (SVD) $X = U\Sigma V^T$ of $X \in \mathbb{R}^{n \times m}$, and $\mathrm{sgn}_+ : \mathbb{R} \to \{0, 1\}$ is the one-sided sign function. As the singular values vary continuously under perturbations in X [4], it is clear the discontinuity in (3.10) manifests from the one-sided sign function—thus, a natural approach to relaxing (3.10) is to first relax the sgn+ function.

Our approach follows the seminal work of Mangasarian et al. [23]. Let $p : \mathbb{R}_+ \to \mathbb{R}_+$ denote a continuous density function and $\nu : \mathbb{R}_+ \to \mathbb{R}_+$ is a continuous increasing function satisfying $\nu(0) = 0$. One way to approximate the sgn_+ function is to integrate τ-smoothed variations $\hat{\delta}$ of the Dirac delta measure δ:

$$(\,\forall\, z \geq 0, \tau > 0\,) \qquad \phi(x, \tau) \triangleq \int_{-\infty}^{x} \hat{\delta}(z, \tau)\, dz, \qquad \hat{\delta}(z, \tau) = \frac{1}{\nu(\tau)} \cdot p\left(\frac{z}{\nu(\tau)}\right) \tag{3.11}$$

In contrast to the sgn_+ function, if p is continuous on $\mathbb{R}_+$ then $\phi(\cdot, \tau)$ is continuously differentiable on $\mathbb{R}_+$, and if p is bounded above on $\mathbb{R}_+$, then $\phi(\cdot, \tau)$ is globally Lipshitz continuous on $\mathbb{R}_+$. Moreover, varying $\tau \in \mathbb{R}_+$ in (3.11) yields an τ-parameterized family of continuous sgn_+ relaxations $\phi : \mathbb{R}_+ \times \mathbb{R}_{++} \to \mathbb{R}_+$, where $\tau > 0$ controls the accuracy of the relaxation.

Many properties of the sign approximation from (3.11) extend naturally to the rank function when substituted appropriately via (3.10). In particular, pairing $X = U\Sigma V^T$ with a scalar-valued ϕ that is continuously differentiable at every entry σ of Σ yields a corresponding *Löwner operator* Φ_τ [5]:

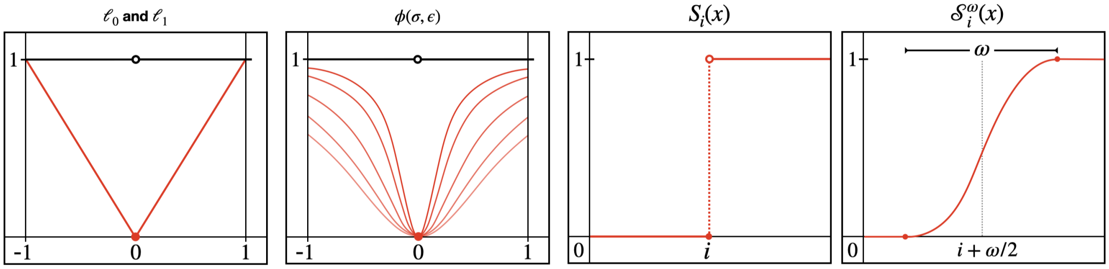

Figure 2: From left to right—the ℓ_1 norm (red) forms a convex envelope over the ℓ_0 (black) pseudo-norm on the interval $[-1, 1]$; $\phi(\cdot, \tau)$ from (3.14) at various values of $\tau > 0$ (red) and at $\tau = 0$ (black); the step function $S_i(x)$ from (3.4); the smoothstep relaxation $\mathcal{S}_i^{\omega}$ from (**??**).

Definition 1 (Spectral ϕ-approximation)**.** Given $X \in \mathbb{R}^{n \times m}$ with SVD $X = U\Sigma V^T$, a fixed $\tau > 0$, and any choice of $\phi : \mathbb{R}_+ \times \mathbb{R}_{++}$ satisfying (3.11), define the *spectral ϕ-approximation* $\Phi_\tau(X)$ of X as:

$$\Phi_\tau(X) \triangleq \sum_{i=1}^{n} \phi(\sigma_i, \tau) u_i v_i^T \tag{3.12}$$

where u_i and v_i are the ith columns of U and V, respectively.

At a high level, Definition (3.12) is closely related to that of a *matrix function* $f(A) \triangleq Uf(\Lambda)U^T$ defined over square matrices [4]. By imposing additional restrictions via (3.11) on ϕ, the operator Φ_τ exhibits a variety of attractive properties related to rank-approximation, monotonicity, and differentiability.

Proposition 2 (Bi et al. [5])**.** *The operator $\Phi_\tau : \mathbb{R}^{n \times m} \to \mathbb{R}^{n \times m}$ defined by* (3.12) *satisfies:*

1. *For any $\tau \geq 0$, the Schatten-1 norm $\|\Phi_\tau(X)\|_*$ of $\Phi_\tau(X)$ is given by $\sum\limits_{i=1}^{n} \phi(\sigma_i, \tau)$*

2. *For any $\tau' \geq \tau$, $\|\Phi_{\tau'}(X)\|_* \leq \|\Phi_\tau(X)\|_*$ for all $X \in \mathbb{R}^{n \times m}$.*

3. *For any given $X \in \mathbb{R}^{n \times m}$ with rank $r = \operatorname{rank}(X)$ and positive singular values $\Lambda(X) = \{\sigma_1, \sigma_2, \ldots, \sigma_r\}$:*

$$0 \leq r - \|\Phi_\tau(X)\|_* \leq r \cdot (1 - \phi(\sigma_r, \tau))$$

Moreover, if τ satisfies $0 < \tau \leq \sigma_r / r$, then $r - \|\Phi_\tau(X)\|_*$ is bounded above by a constant $c_\phi(r) \geq 0$.

4. *$\|\Phi_\tau(X)\|_*$ is globally Lipshitz continuous and semismooth*[3] *on $\mathbb{R}^{n \times m}$.*

Noting property (4), since the sum Lipshitz functions is also Lipshitz, it is easy to verify that replacing the rank function in all of the constitutive terms from Proposition 1 yields Lipshitz continuous functions whenever the filter function f_α is itself Lipshitz and the step functions from (3.4) are smoothed ($\omega > 0$).

Remark 2. Though Φ_τ is a continuously differentiable operator[4] in $\mathbb{R}^{n \times m}$ for any $\tau > 0$, its Schatten-1 norm $\|\Phi_\tau(X)\|_*$ is only directionally differentiable everywhere on $\mathbb{R}^{n \times m}$ in the Hadamard sense, due to Proposition 2.2(d-e) of [5]. However, $\|\Phi_\tau(X)\|_*$ is differentiable on the positive semi-definite cone $\mathbb{S}_+^n$.

Interpretation #1: In sparse inverse applications, it is commonplace to regularize an objective function to prevent the problem from being ill-posed. For example, the classical least-squares approach to solving the linear system $Ax = b$ is often augmented with the *Tikhonov regularization* (TR) for some $\tau > 0$:

$$x_\tau^* = \operatorname*{arg\,min}_{x \in \mathbb{R}^n} \|Ax - b\|^2 + \tau\|x\|^2 = (A^T A + \tau I)^{-1} A^T b \tag{3.13}$$

[3] Here, "semismooth" refers to the existence certain directional derivatives in the limit as $\tau \to 0^+$, see [4, 5].

[4] In fact, it may be shown to be twice continuously differentiable at X if ϕ is twice-differentiable at each $\sigma_i(X)$, see [15])

When $\tau = 0$, one recovers the standard ℓ_2 minimization, whereas when $\tau > 0$ solutions x^*_τ with small norm are favored. Similarly, by parameterizing ϕ by $\nu(\tau) = \sqrt{\tau}$ and $p(x) = 2x(x^2 + 1)^{-2}$, one obtains via (3.11):

$$\phi(x, \tau) = \int_0^z \hat{\delta}(z, \tau) dz = \frac{2}{\tau} \int_0^z z \cdot \left((z/\sqrt{\tau})^2 + 1 \right)^{-2} dz = \frac{x^2}{x^2 + \tau} \tag{3.14}$$

By substituting $\mathrm{sgn}_+ \mapsto \phi$ and composing with the singular value function (3.12), the corresponding spectral rank approximation reduces[5] to the following *trace* formulation:

$$\|\Phi_\tau(A)\|_* = \sum_{i=1}^{n} \frac{\sigma_i(A)^2}{\sigma_i(A)^2 + \tau} = \mathrm{Tr}\left[(A^T A + \tau\, I)^{-1} A^T A \right] \tag{3.15}$$

The relaxation level τ may be thought of as a regularization parameter that preferences smaller singular values: larger values smooth out $\|\Phi_\tau(\cdot)\|_*$ by making the pseudo-inverse less sensitive to perturbations, whereas smaller values lead to a more faithful[6] approximations of the rank. In this sense, we interpret the quantities obtained by applying (3.12) to the terms from Proposition 1 as *regularized RI approximation.*

Interpretation #2: In shape analysis applications, matrix functions are often used to simulate diffusion processes on meshes or graphs embedded in $\mathbb{R}^d$ to obtain information of about their geometry. For example, consider a weighted graph $G = (V, E)$ with $n = |V|$ vertices with graph Laplacian $L_G = \partial_1 \partial_1^T$. The *heat* of every vertex $v(t) \in \mathbb{R}^n$ as a function of time $t \geq 0$ is governed by L_G and the *heat equation* []:

$$v'(t) = -L_G v(0) \quad \iff \quad L_G \cdot u(x, t) = -\partial u(x, t)/\partial t \tag{3.16}$$

To simulate a diffusion process on G from an initial distribution of heat $v(0) \in \mathbb{R}^n$, it suffices to construct the *heat kernel* $H_t \triangleq \exp(-t \cdot L_G)$ via the spectral decomposition $L_G = U \Lambda U^T$ of L_G:

$$v(t) = H_t\, v(0), \text{ where } H_t = \sum_{i=1}^{n} e^{-t\lambda_i}\, u_i\, u_i^T \tag{3.17}$$

The heat kernel is invariant under isometric deformations, stable under perturbations, and is known to contain multiscale geometric information due to its close connection to geodesics []. As is clear from (3.17), it is also a matrix function. Now, consider (3.11) with $\nu(\tau) = \tau$ and $p(\lambda) = \exp(-\lambda_+)$ where $x_+ = \max(x, 0)$:

$$\phi(\lambda, \tau) = \int_0^z \hat{\delta}(z, \tau) dz = \frac{1}{\tau} \int_0^z \exp(-z/\tau) dz = 1 - \exp(-\lambda/\tau), \qquad \text{for all } \lambda \geq 0 \tag{3.18}$$

In the context of diffusion, observe the parameter τ is inversely related diffusion time (i.e. $t = 1/\tau$) and that as $t \to 0$ (or $\tau \to \infty$) the expression $1 - \exp(-\lambda/\tau)$ approaches the sgn_+ function on the interval $[0, \infty)$. As above, substituting ϕ appropriately into Definition (3.12) again yields an equivalent trace expression:

$$\|\Phi_\tau(L_G)\|_* = \sum_{i=1}^{n} 1 - \exp(-\lambda_i/\tau) = n - \mathrm{Tr}\left[H_{1/\tau} \right] \tag{3.19}$$

The heat kernel H_t has been shown to fully characterize shapes up to isometry, motivating the creation of various geometric signatures, such as the Heat Kernel Signature (HKS) [] and the Heat Kernel Trace []. In this sense, we interpret our spectral rank relaxation using (3.18) as a *geometrically informative RI approximation.*

[5] See Theorem 2 of [35] for a proof of the second equality.

[6] This can be seen directly by (3.13) as well, wherein increasing τ lowers the condition number of $A^T A + \tau I$ monotonically, signaling a tradeoff in stability at the expense of accuracy.

4 Computational Implications

Exact computation

Computing the spectral quantities $\hat{\mu}_p^*$ and $\hat{\beta}_p^*$ reduces to computing eigenvalues of Laplacian operators. To do this, we employ the *Lanczos method* [22], which estimates the eigenvalues of any symmetric linear operator A via projection onto successive Krylov subspaces. Formally, given a symmetric $A \in \mathbb{R}^{n \times n}$ with eigenvalues $\lambda_1 \geq \lambda_2 > \cdots \geq \lambda_r > 0$ and a vector $v \neq 0$, the Lanczos method generates the following triplet of matrices:

$$\begin{aligned} K &= [\, A^0 v \mid A^1 v \mid A^2 v \mid \cdots \mid A^{r-1} v \,] \\ Q &= [\, q_1, q_2, \ldots, q_r \,] \leftarrow \mathrm{qr}(K) \\ T &= Q^T A Q \end{aligned}$$

where $K \in \mathbb{R}^{n \times r}$ is the *Krylov matrix* with respect to (A, v), $Q \in \mathbb{R}^{n \times r}$ is an orthogonal change-of-basis, and $T \in \mathbb{R}^{r \times r}$ is symmetric tridiagonal *Jacobi matrix.* It is well known that Q is a similarity transform, i.e. T preserves the spectrum of A and the problem of finding eigenvalues reduces to diagonalizing T.

The Lanczos method is often called a "matrix free" method due to the fact its only requirement to execute is a matrix-vector product operator $v \mapsto Av$, implying A need not necessarily be stored in memory explicitly. Indeed, due to the *three-term recurrence* [33], execution of the Lanczos method requires just three $O(n)$-sized vectors and a few $O(n)$ vector operations—neither K nor Q need be formed explicitly.

Lemma 2 ([27])**.** *Given a symmetric rank-r matrix $A \in \mathbb{R}^{n \times n}$ whose matrix-vector operator $A \mapsto Ax$ requires $O(\eta)$ time and $O(\nu)$ space, the Lanczos iteration computes $\Lambda(A) = \{\lambda_1, \lambda_2, \ldots, \lambda_r\}$ in $O(\max\{\eta, n\} \cdot r)$ time and $O(\max\{\nu, n\})$ space, when computation is done in exact arithmetic.*

As evidenced by Lemma 2, the efficiency of the Lanczos method depends on the availability of a fast `matvec` operator, such as those arising from structured or sparse[7] operators. An exemplary structured operator is the graph Laplacian $L = \partial_1 \partial_1^T$, whose $x \mapsto Lx$ operation has complexity linear in the number of edges $|E|$ due to its graph structure. Though this is well established result for the graph Laplacian (i.e. the case $p = 1$), it is not immediately clear whether a similar result generalizes to combinatorial Laplacian operators derived from simplicial complexes—our next result affirms this.

Lemma 3. *For any $p \geq 0$ and simplicial complex K with $n = |K^p|$ and $m = |K^{p+1}|$, if there exists a hash function $h : K^p \to [n]$ with $O(1)$ access time and $O(c)$ storage, then there exists a two-phase algorithm for computing the inner product $x \mapsto \langle\, L_p, x \,\rangle$ in $O(m(p+1))$ time and $O(\max(c, m))$ storage.*

The algorithm and proof are given in appendix section A.1. From a practical perspective, many hash table implementations achieve expected $O(1)$ access time using only a linear amount of storage, and as $p \geq 0$ is typically quite small—typically no greater than two—the operation $x \mapsto Lx$ in practice exhibits $\approx O(m)$ time and space complexities. We delegate more practical issues regarding the computation to appendix A.1. Combining Lemmas 2 and 3 yields our main result.

Proposition 3. *For any constant $p \geq 0$ and box $R = [a, b] \times [c, d] \subset \Delta_+$, the persistent rank-based quantities $\mu_p^R(K)$ and $\beta_p^{a,b}(K)$ derived from a simplicial complex K with $n = |K^p|$ and $m = |K^{p+1}|$ simplices can be computed in:*

$$\mu_p^R(K) \sim O(m \cdot)$$

In particular, for a given box , $\mu_p^R(K)$ has complexity $O(c^2)$ where $y_{ad} = |K_d^{p+1}| - |K_a^{p+1}|$

Corollary 3. *For any constant $p \geq 0$, the persistent rank-based quantities $\mu_p^*(K)$ and $\beta_p^*(K)$ derived from a simplicial complex K with $n = |K^p|$ and $m = |K^{p+1}|$ simplices can be computed in $O(mr)$ time and $O(m)$ storage via the Lanczos iteration, when carried out in exact arithmetic.*

[7] For example, the Lanczos method has expected time complexity $O(nzr)$ for sparse matrices containing an average of z nonzeros per row [19]

Remark 3. The standard reduction-family of algorithms computes the p-th persistent homology of a filtration K of size $N = |K|$ in $O(N^3)$ time and $O(N^2)$ space, respectively (these bounds are actually tight $\Theta(N^3)$, see []). Interestingly, Chen and Kerber [10] have shown that since the persistence diagram contains at most $N/2 = O(N)$ points, it may be constructed using at most $2N - 1$ "μ-queries" (evaluations of μ_p^R) via a divide-and-conquer scheme on the index-persistence plane—thus, by Theorem 3, we recover the $O(N^3)$ complexity of the reduction algorithm using only rank computations.

Randomized (η, ϵ)-approximation via trace estimation

As in [27], Proposition 3 assumes an exact arithmetic computation model to simplify both the presentation of the theory and the corresponding complexity statements. In practice, finite-precision arithmetic introduces both rounding and cancellation errors into the computation, which primarily manifests as loss of orthogonality between the Lanczos vectors. These errors not only affect the convergence rate and termination of Lanczos, but often cause a host of other issues [27], prohibiting any practical use of the original Lanczos algorithm. Though many improvements have been proposed (i.e. selective re-orthogonalization or implicit restarts), it is well-known that obtaining accurate eigenvalue approximations on the interior of the spectrum is quite expensive with the Lanczos method.

Fortunately, it turns out estimating eigenvalues accurately is not necessary for accurate rank estimation. It has long been known that empirically the Lanczos method can be used to approximate the action $x \mapsto f(A)x$, where $f(A) = Uf(\Lambda)U^T$ represents a matrix function[8] defined for the pair (A, f):

$$Q^T AQ = T \quad \Leftrightarrow \quad f(A)v \approx \|x\| \cdot Qf(T)e_1$$

Unlike eigenvalue estimation, it has been shown that using Lanczos in finite precision arithmetic is actually stable for matrix function approximation; in particular, Paige's A27 Lanczos variant [] executed up to degree k is no worse than the best degree-p polynomial approximation to f, for any $p < k$. For general matrix functions, this implies that finite-precision Lanczos essentially matches the strongest known exact arithmetic bounds.

The significance of the above finite-precision results is best illustrated by the *stochastic Lanczos quadrature* method employed by Ubaru et. al [] for numerical rank estimation, which estimates the trace of an arbitrary matrix function by means of a Girard-Hutchinson (GH) estimator of the trace:

$$\mathrm{tr}(f(A)) \approx \frac{n}{n_v} \sum_{i=1}^{n_v} v_i^T f(A) v_i$$

Using Lanczos quadrature estimates, the theory above enables the GH estimator to extend to be used as an estimator for the trace of a matrix function:

$$\mathrm{tr}(f(A)) \approx \frac{n}{n_v} \sum_{i=1}^{n_v} e_1^T f(T_m) e_1 = \frac{n}{n_v} \sum_{j=1}^{n_v} \left(\sum_{i=1}^{m} \tau_i^{(j)} f(\theta_i)^{(j)} \right)$$

Under relatively mild assumptions, if the function of interest $f : [a, b] \to \mathbb{R}$ is analytic on $[\lambda_{\min}, \lambda_{\max}]$, then for constants $\epsilon, \eta \in (0, 1)$ the output Γ of the GH estimator satisfies:

$$\Pr\left[|\mathrm{tr}(f(A)) - \Gamma| \leq \epsilon |\mathrm{tr}(f(A))| \right] \geq 1 - \eta$$

In other words, we can achieve a relative ϵ-approximation of $\mathrm{tr}(f(A))$ with success probability η using on the order of $\sim O(\epsilon^{-2} \log(\eta^{-1}))$ evaluations of $e_1^T f(T_m) e_1$. This probablistic guarantee is most useful when ϵ is not too small, i.e. only a relatively coarse approximation of $\mathrm{tr}(f(A))$ is needed. More recent results by Musco et al. [] improve upon this by showing how to extend the GH algorithm to obtain a $(1 \pm \epsilon)$ approximation using only $\sim O(\epsilon^{-1} \log(\eta^{-1}))$ samples, though at a small additional re-orthogonalization and storage costs.

[8] Recall a matrix function...

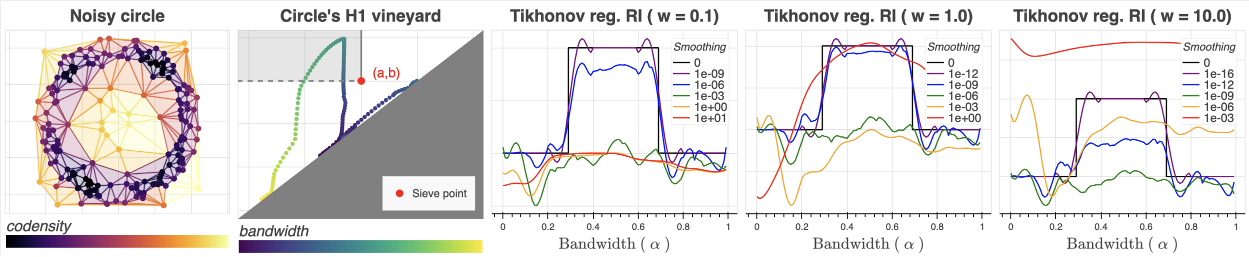

Figure 3: From left to right: Delaunay complex K realized from point set $X \subset \mathbb{R}^2$ sampled with multiple types of noise around S^1 (colored by codensity at optimal $\alpha^* \approx 1/2$); codensity vineyard of (K, f_α) across varying bandwidths α and a fixed sieve point $(a, b) \in \Delta_+$; Tikhonov regularizations $\hat{\beta}_p^{a,b}(\alpha)$ at varying regularization (τ) and sign width (ω) values. Observe lower values of τ lead to approximations closer to the rank (black) at the cost of smoothness, while larger values can yield very smooth albeit possibly uninformative relaxations.

5 Applications & Experiments

Filtration optimization

It is common in TDA for the filter function $f : K \to \mathbb{R}$ to depend on hyper-parameters. For example, prior employing persistence, it is common to remove outliers from point set $X \subset \mathbb{R}^d$ using a density-based pruning heuristic that itself is parameterized. Though stable under Hausdorff noise [], diagrams are notably unstable against *strong outliers*—these outliers can obscure the detection of non-trivial topological space. Even one point can ruin the summary. As an exemplary use-case of our spectral-based method, below we re-cast the problem of identifying strong outliers as a problem of *filtration optimization*.

Consider a Delaunay complex K realized from a point set $X \subset \mathbb{R}^2$ sampled around S^1 affected by both Hausdorff noise and strong outliers, shown in Figure 3. One approach to detect the presence of S^1 in the presence of such outliers is maximize $\beta_p^{a,b}(\alpha)$ for some fixed $(a, b) \in \Delta_+$ over the pair (K, f_α), where $f_\alpha : X \to \mathbb{R}_+$ is a bandwidth-parameterized kernel (co)-density estimate:

$$\alpha^* = \underset{\alpha \in \mathbb{R}}{\arg\max}\ \beta_p^{a,b}(K, f_\alpha), \qquad \text{where } f_\alpha(x) = \frac{1}{n\alpha} \sum_i C(\mathcal{K}) - \mathcal{K}_\alpha(x_i - x) \tag{5.1}$$

where $C(\mathcal{K})$ is a normalizing constant that depends on $\mathcal{K}$. The intuition is that there is likely some choice of bandwidth α^* which separates the strong outliers from the densely packed points clustered around S^1. Thus, if (a, b) is appropriately chosen, we expect $\beta_p^{a,b}(\alpha) = 1$ at the optimal bandwidth α^* (matching the first Betti number of S^1) and 0 otherwise for bandwidth choices that cannot distinguish the points packed densely around S^1 against the strong outliers.

The behavior is reflected by the vineyard of $\mathrm{dgm}_1(K, f_\alpha)$, which shown on the second plot of Figure 3; the sieve point (a, b) and the region it corresponds too is also shown. As $\beta_p^{a,b}$ is an integer-valued invariant, it is discontinuous and difficult to optimize; in contrast, we know from Proposition (2) that we can obtain a continuous and differentiable relaxation of $\beta_p^{a,b}$ by replacing $\beta_p^{a,b} \mapsto \hat{\beta}_p^{a,b}$ in (5.1), enabling the use of first-order optimization techniques. Moreover, the connection to the Tikhonov regularization from **??** and Lemma **??** suggests we various iterative optimization methods... One such optimization technique is the iterative sparse thresholding method, which alternates between linearization and regularization steps to solve ill-posed or poorly behaved optimization problems []. We leave the tuning of such optimizers to future work.

Manifold detection from image patches

Topology-guided simplification

In many 3D computer graphics applications, one typically has access to a simplicial or polygonal mesh embedded in $\mathbb{R}^3$ that would like to simplify to decrease its level of detail (LOD). Such simplifications are

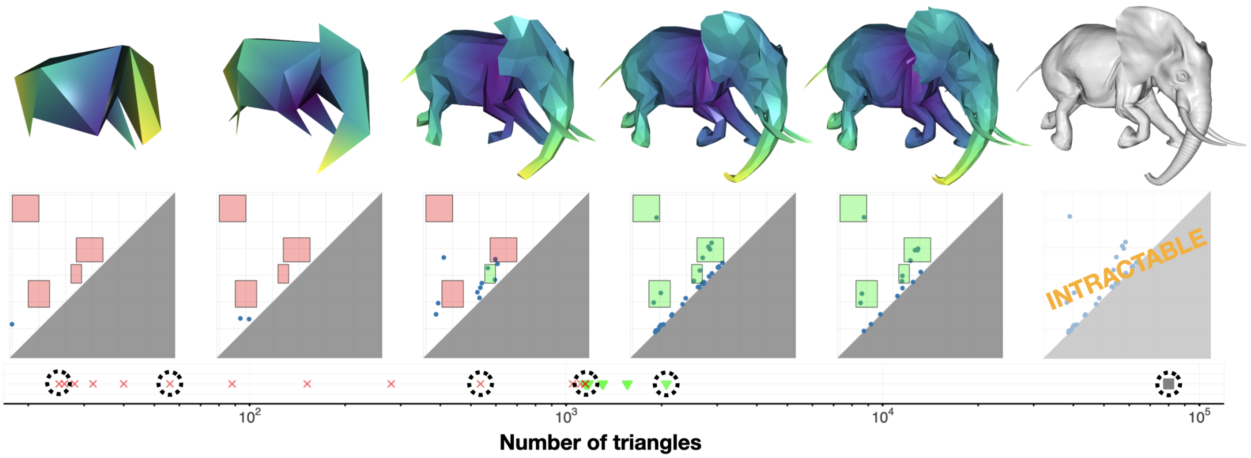

Figure 4: (Top) Meshes filtered and colored by eccentricity at varying levels of simplification; (middle) their diagrams and topological constraints; (bottom) simplification thresholds tested by an exponential search, on a logarithmic scale. The color/shape of the markers indicate whether the corresponding meshes meet (green triangle) or do not meet (red x) the topological constraints of the sieve—the gray marker identifies the original mesh (not used in the search). Black dashed circles correspond with the meshes in the top row.

often necessary to improve the efficiency of compute-intensive tasks that depend on the size of the mesh (e.g. rendering). Methods for simplifying meshes based on the preservation of various geometrical properties, such as curvature, co-planarity, or distance-based functions of the shape are well-known, see e.g. facet merging, face decimation and energy minimization in [] for an overview.

In many applications, simplifications defined solely using geometric properties is not enough—often, such simplifications are preferred to be *topology-preserving* in some sense, e.g. edge collapses must preserve neighboring connectivity or the number of holes. Though such operations are often critical to..., they are inherently *local* conditions. With the exception of very simple properties (e.g. connectedness, the Euler characteristic, etc.) it is often far too expensive to ensure *global* topological properties of shapes are preserved under incremental simplification procedures, either because the properties themselves are expensive to check (per simplifying operation) or because the existing simplification procedure cannot ensure such guarantees.

To illustrate another use-case of our methodology, we look to address the problem of mesh simplification under the preservations of persistence-based measures of topology. Consider a 3D mesh library of animals pose data... As these are high quality meshes, one may be interested in geometric simplifications, such as quadric decimation procedure, which takes as input the number of triangles n_2 desired in the output. A priori, it is unclear how to set k—however, since the taxonomical classification of these meshes are known, one simple criterion for look to preserve the is appearance of the major extremities, such as limbs and tails.

6 Conclusion & Future Work

Interestingly, our results also imply the existence of an efficient output-sensitive algorithm for computing Γ-persistence pairs with at least $(\Gamma > 0)$-persistence (via [10]) that requires the operator $x \mapsto \partial x$ as its only input, which we consider to be of independent interest.

References

[1] Henry Adams, Tegan Emerson, Michael Kirby, Rachel Neville, Chris Peterson, Patrick Shipman, Sofya Chepushtanova, Eric Hanson, Francis Motta, and Lori Ziegelmeier. Persistence images: A stable vector representation of persistent homology. *Journal of Machine Learning Research*, 18, 2017.

[2] Ulrich Bauer and Michael Lesnick. Persistence diagrams as diagrams: A categorification of the stability theorem. In *Topological Data Analysis*, pages 67–96. Springer, 2020.

[3] Ulrich Bauer, Talha Bin Masood, Barbara Giunti, Guillaume Houry, Michael Kerber, and Abhishek Rathod. Keeping it sparse: Computing persistent homology revised. *arXiv preprint arXiv:2211.09075*, 2022.

[4] Rajendra Bhatia. *Matrix analysis*, volume 169. Springer Science & Business Media, 2013.

[5] Shujun Bi, Le Han, and Shaohua Pan. Approximation of rank function and its application to the nearest low-rank correlation matrix. *Journal of Global Optimization*, 57(4):1113–1137, 2013.

[6] Peter Bubenik et al. Statistical topological data analysis using persistence landscapes. *J. Mach. Learn. Res.*, 16(1):77–102, 2015.

[7] Andrea Cerri, Barbara Di Fabio, Massimo Ferri, Patrizio Frosini, and Claudia Landi. Betti numbers in multidimensional persistent homology are stable functions. *Mathematical Methods in the Applied Sciences*, 36(12):1543–1557, 2013.

[8] Frédéric Chazal, David Cohen-Steiner, Leonidas J Guibas, Facundo Mémoli, and Steve Y Oudot. Gromov-hausdorff stable signatures for shapes using persistence. In *Computer Graphics Forum*, volume 28, pages 1393–1403. Wiley Online Library, 2009.

[9] Frédéric Chazal, Vin De Silva, Marc Glisse, and Steve Oudot. *The structure and stability of persistence modules*, volume 10. Springer, 2016.

[10] Chao Chen and Michael Kerber. An output-sensitive algorithm for persistent homology. In *Proceedings of the twenty-seventh annual symposium on Computational geometry*, pages 207–216, 2011.

[11] Fan RK Chung. *Spectral graph theory*, volume 92. American Mathematical Soc., 1997.

[12] David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. Stability of persistence diagrams. In *Proceedings of the twenty-first annual symposium on Computational geometry*, pages 263–271, 2005.

[13] David Cohen-Steiner, Herbert Edelsbrunner, and Dmitriy Morozov. Vines and vineyards by updating persistence in linear time. In *Proceedings of the twenty-second annual symposium on Computational geometry*, pages 119–126, 2006.

[14] Tamal Krishna Dey and Yusu Wang. *Computational topology for data analysis.* Cambridge University Press, 2022.

[15] Chao Ding, Defeng Sun, Jie Sun, and Kim-Chuan Toh. Spectral operators of matrices. *Mathematical Programming*, 168(1):509–531, 2018.

[16] Herbert Edelsbrunner and John L Harer. *Computational topology: an introduction.* American Mathematical Society, 2022.

[17] Herbert Edelsbrunner, David Letscher, and Afra Zomorodian. Topological persistence and simplification. In *Proceedings 41st annual symposium on foundations of computer science*, pages 454–463. IEEE, 2000.

[18] Timothy E Goldberg. Combinatorial laplacians of simplicial complexes. *Senior Thesis, Bard College*, 6, 2002.

[19] Gene H Golub and Charles F Van Loan. *Matrix computations.* JHU press, 2013.

[20] Danijela Horak and Jürgen Jost. Spectra of combinatorial laplace operators on simplicial complexes. *Advances in Mathematics*, 244:303–336, 2013.

[21] Woojin Kim and Facundo Mémoli. Spatiotemporal persistent homology for dynamic metric spaces. *Discrete & Computational Geometry*, 66:831–875, 2021.

[22] Cornelius Lanczos. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. 1950.

[23] Olvi Mangasarian and Chunhui Chen. A class of smoothing functions for nonlinear and mixed complementarity problems. Technical report, 1994.

[24] Facundo Mémoli, Zhengchao Wan, and Yusu Wang. Persistent laplacians: Properties, algorithms and implications. *SIAM Journal on Mathematics of Data Science*, 4(2):858–884, 2022.

[25] Michael William Newman. The laplacian spectrum of graphs. Master's thesis, 2001.

[26] Arnur Nigmetov and Dmitriy Morozov. Topological optimization with big steps. *arXiv preprint arXiv:2203.16748*, 2022.

[27] Beresford N Parlett. Do we fully understand the symmetric lanczos algorithm yet. *Brown et al*, 3:93–107, 1994.

[28] Jose A Perea. Persistent homology of toroidal sliding window embeddings. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6435–6439. IEEE, 2016.

[29] Jose A Perea, Elizabeth Munch, and Firas A Khasawneh. Approximating continuous functions on persistence diagrams using template functions. *Foundations of Computational Mathematics*, pages 1–58, 2022.

[30] Matthew Piekenbrock and Jose A Perea. Move schedules: Fast persistence computations in coarse dynamic settings. *arXiv preprint arXiv:2104.12285*, 2021.

[31] Chi Seng Pun, Kelin Xia, and Si Xian Lee. Persistent-homology-based machine learning and its applications–a survey. *arXiv preprint arXiv:1811.00252*, 2018.

[32] Luis Scoccola and Jose A Perea. Fibered: Fiberwise dimensionality reduction of topologically complex data with vector bundles. In *39th International Symposium on Computational Geometry (SoCG 2023)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2023.

[33] Horst D Simon. Analysis of the symmetric lanczos algorithm with reorthogonalization methods. *Linear algebra and its applications*, 61:101–131, 1984.

[34] Katharine Turner, Sayan Mukherjee, and Doug M Boyer. Persistent homology transform for modeling shapes and surfaces. *Information and Inference: A Journal of the IMA*, 3(4):310–344, 2014.

[35] Yun-Bin Zhao. An approximation theory of matrix rank minimization and its application to quadratic equations. *Linear Algebra and its Applications*, 437(1):77–93, 2012.

[36] Afra Zomorodian and Gunnar Carlsson. Computing persistent homology. In *Proceedings of the twentieth annual symposium on Computational geometry*, pages 347–356, 2004.

A Appendix

Expanded Intro

Though homology is primarily studied as a topological invariant, the fact that persistent homology encodes both topological and geometric information in its diagram has motivated its use not only as a shape descriptor but also as a metric invariant. Metric invariants, or "signatures," are commonly used in metric learning to ascertain whether two comparable data sets X, Y represent the same object—typically up to a some notion of invariance. One mathematically attractive model for measuring the dissimilarity between shapes/datasets is the Gromov-Hausdorff (GH) distance $d_{\mathrm{GH}}((X, d_X), (Y, d_Y)$ between compact metric spaces $(\mathcal{X}, d_X), (\mathcal{Y}, d_Y)$: by altering the choice of metric (d_X, d_Y), the corresponding metric-distance d_{GH} can be adapted to a chosen notion of invariance [] or to increase its discriminating power []. Though it is NP-hard to compute [], the GH distance defines a metric on the set of isomorphism classes of compact metric spaces endowed with continuous real-valued functions, justifying its study as a mathematical model for shape matching and metric learning. Moreover, it is known that the GH distance is tightly lower-bounded by the bottleneck distance between persistence diagrams constructed over Rips filtrations $R(X, d_X), R(Y, d_Y)$ [], which can be computed in polynomial time. Indeed, Solomon et al [] showed distributed persistence invariants characterize the quasi-isometry type of the underlying space, allowing one to provably interpolate between geometric and topological structure.

Though theoretically well-founded and information dense, persistence diagrams come with their own host of practical issues: they are sensitive to strong outliers, far from injective, and their de-facto standard computation exhibits high algorithmic complexity. Moreover, the space of persistence diagrams $\mathcal{D}$ is a Banach space, preventing one from doing even basic statistical operations, such as averaging []. As a result, many researchers have focused on extending, enhancing, or otherwise supplementing persistence diagrams with additional information. Turner et al [] proposed associating a collection a shape descriptors with a PL embedded $X \subset \mathbb{R}^d$—one descriptor for each point on S^{d-1}—which they called a *transform.* More exactly, suppose both the data X and its geometric realization K are PL embedded in $\mathbb{R}^d$ and has centered and scaled appropriately. The main theorem in [] is that associating a persistence diagram, or even a simpler descriptor such as the Euler characteristic, for every point on S^{d-1} is actually sufficient information to theoretically reconstruct K.

Missing from the above work is the are two important directions: how do you configure such transforms to retain the important topological/geometric information and discard irrelevant information, and (2) how may we efficiently compute them? The former question is synonymous with choosing the invariance model in the GH framework, which seems to be highly domain specific. In the latter case, though we know the number of directions is bounded [], the bound is simply too high to be of any practical use. While there are efficient algorithms for both the ECC and persistence computations in static settings, the state of the art in parameterized settings is non-trivial and ongoing research area.

Expanded Background

Laplacian Energy: Ever since Kirchoff's matrix tree theorem, which relates any cofactor of the graph Laplacian to the number of spanning trees of a graph.... functions summarizing the spectra of Laplacian operators with a scalar value have found many applications, from quantifying hierarchical image complexities, to summarizing electrical resistance between vertices in a circuit network, to indicating the melting or boiling point of certain polycyclic aromatic hydrocarbons in chemical applications []. More generally, the sum of the largest k eigenvalues of L is related to the clique number of the graph, as a measure of complexity. is often termed the *Laplacian energy*, has used

A.1 Combinatorial Laplacians

The natural extension of the graph Laplacian L to simplicial complexes is the *p-th combinatorial Laplacian* Δ_p, whose explicit matrix representation is given by:

$$\Delta_p(K) = \underbrace{\partial_{p+1} \circ \partial_{p+1}^T}_{L_p^{\mathrm{up}}} + \underbrace{\partial_p^T \circ \partial_p}_{L_p^{\mathrm{dn}}} \tag{A.1}$$

Indeed, when $p = 0$, $\Delta_0(K) = \partial_1 \partial_1^T = L$ recovers the graph Laplacian. As with boundary operators, $\Delta_p(K)$ encodes simplicial homology groups in its nullspace, a result known as the discrete Hodge Theorem []:

$$\tilde{H}_p(K; \mathbb{R}) \cong \ker(\Delta_p(K)), \quad \beta_p = \mathrm{nullity}(\Delta_p(K)) \tag{A.2}$$

The fact that the Betti numbers of K may be recovered via the nullity of $\Delta_p(K)$ has been well studied (see e.g. Proposition 2.2 of []). In fact, as pointed out by [], one need not only consider Δ_p as the spectra of Δ_p, L_p^{up}, and L_p^{dn} are intrinsically related by the identities:

$$\Lambda(\Delta_p(K)) \doteq \Lambda(L_p^{\mathrm{up}}) \,\dot{\cup}\, \Lambda(L_p^{\mathrm{dn}}), \qquad \Lambda(L_p^{\mathrm{up}}) \doteq \Lambda(L_{p+1}^{\mathrm{dn}}) \tag{A.3}$$

where $A \doteq B$ and $A \,\dot{\cup}\, B$ denotes equivalence and union between the *non-zero* elements of the multisets A and B, respectively. Moreover, all three operators Δ_p, L_p^{up}, and L_p^{dn} are symmetric, positive semidefinite, and compact—thus, for the purpose of estimating β_p, it suffices to consider only one family of operators.

To translate the continuity results from definition **??** to any of the Laplacian operators above, we must consider weighted versions. Here, a *weight function* is a non-negative real-valued function defined over the set of all faces of K:

$$w : K \to \mathbb{R}_+ \tag{A.4}$$

The set of weight functions and the choice of scalar product on $C^p(K, \mathbb{R})$ wherein elementary cochains are orthogonal are in one-to-one correspondence [] (see Appendix **??**). In this way, we say that the weight function *induces* an inner product on $C^p(K, \mathbb{R})$:

$$\langle\, f, g \,\rangle_w = \sum_{\sigma \in K^p} w(\sigma) f([\sigma]) g([\sigma]) \tag{A.5}$$

Moreover, Laplacian operators are uniquely determined by the choice of weight function. This correspondence permits us to write the matrix representation of Δ_p explicitly:

$$\Delta_p(K, w) \triangleq W_p^+ \partial_{p+1} W_{p+1} \partial_{p+1}^T \;+\; \partial_p^T W_p^+ \partial_p W_{p+1} \tag{A.6}$$

where $W_p = \mathrm{diag}(\{\, w(\sigma_i) \,\}_{i=1}^n)$ represents a non-negative diagonal matrices restricted $\sigma \in K^p$ and W^+ denotes the pseudoinverse. Note that (A.6) recovers (3.7) in the case where w is the constant map $w(\sigma) = 1$, which we call the *unweighted* case.

Unfortunately, various difficulties arise with weighting combinatorial Laplacians with non-constant weight functions, such as asymmetry, scale-dependence, and spectral instability. Indeed, observe that in general neither terms in (A.6) are symmetric unless $W_p = I_n$ (for L_p^{up}) or $W_{p+1} = I_m$ (for L_p^{dn}). However, as noted in [24], L_p^{up} may be written as follows:

$$L_p^{\mathrm{up}} = W_p^+ \partial_{p+1} W_{p+1} \partial_{p+1}^T = W_p^{+/2} \big(W_p^{+/2} \partial_{p+1} W_{p+1} \partial_{p+1}^T W_p^{+/2} \big) W_p^{1/2} \tag{A.7}$$

Since (A.7) is of the form $W^+ P W$ where $P \in S_n^+$ and W is a non-negative diagonal matrix, this rectifies the symmetry problem. Towards bounding the spectra of L_p^{up}, Horek and Jost [] propose *normalizing* Δ_p by augmenting w's restriction to K^p:

$$w(\tau) = \sum_{\tau \in \partial(\sigma)} w(\sigma) \quad \forall\, \tau \in K^p,\, \sigma \in K^{p+1} \tag{A.8}$$

Substituting the weights of the p-simplices in this way is equivalent to mapping $W_p \mapsto \mathcal{D}_p$ where $\mathcal{D}_p$ is the *diagonal degree matrix.* The corresponding substitution in (A.7) yields the *weighted combinatorial normalized Laplacian* (up-)operator:

$$\mathcal{L}_p^{\mathrm{up}} = (\mathcal{D}_p)^{+/2} \partial_p W_{p+1} \partial_p^T (\mathcal{D}_p)^{+/2} = \mathcal{I}_n - \mathcal{A}_p^{\mathrm{up}} \tag{A.9}$$

where $\mathcal{A}_p^{\mathrm{up}}$ is a weighted adjacency matrix, and $\mathcal{I}_n$ is the identity matrix with $\mathcal{I}(\tau) = \mathrm{sign}(w(\tau))$ (see Section **??**). The primary benefit of this normalization is that it guarantees $\Lambda(\mathcal{L}_p^{\mathrm{up}}) \subseteq [0, p+2]$ for any choice of weight function, from which one obtains several useful implications, such as tight bounds on the spectral norm []. The same results holds for up-, down-, and combinatorial Laplacians. Moreover, as we will show in a subsequent section, one obtains stability properties with degree-normalization not shared otherwise.

Remark 4. Compared to (A.7), is it worth remarking that one important quality lost in preferring $\mathcal{L}_p^{\mathrm{up}}$ over L_p^{up} is diagonal dominance.

Laplacian matvec

We first recall the characteristics of the graph Laplacians $x \mapsto Lx$ operation. Given a simple undirected graph $G = (V, E)$, let $A \in \{0,1\}^{n \times n}$ denote its binary adjacency matrix satisfying $A[i,j] = 1 \Leftrightarrow i \sim j$ if the vertices $v_i, v_j \in V$ are adjacent in G, and let $D = \mathrm{diag}(\{\deg(v_i)\})$ denote the diagonal *degree* matrix, where $\deg(v_i) = \sum_{j \neq i} A[i,j]$. The *graph Laplacian*'s adjacency, incidence, and element-wise definitions are:

$$L = D - A = \partial_1 \circ \partial_1^T , \qquad L\,[i,j] = \begin{cases} \deg(v_i) & \text{if } i = j \\ -1 & \text{if } i \sim j \\ 0 & \text{if } i \nsim j \end{cases} \tag{A.10}$$

Furthermore, by using the adjacency relation $i \sim j$ as in [11], the linear and quadratic forms of L may be succinctly expressed as:

$$(\,\forall x \in \mathbb{R}^n\,) \qquad (Lx)_i = \deg(v_i) \cdot x_i - \sum_{i \sim j} x_j \,, \qquad x^T L x = \sum_{i \sim j} (x_i - x_j)^2 \tag{A.11}$$

If G has m edges and n vertices taking labels in the set $[n]$, computing the product from (A.11) requires just $O(m)$ time and $O(n)$ storage via two edge traversals: one to accumulate vertex degrees and one to remove components from incident edges. By precomputing the degrees, the operation reduces further to a single $O(n)$ product and $O(m)$ edge pass, which is useful when repeated evaluations for varying values of x are necessary.

To extend the two-pass algorithm outlined above when $p > 0$, we first require a generalization of the connected relation from (A.11). Denote with $\mathrm{co}(\tau) = \{\, \sigma \in K^{p+1} \mid \tau \subset \sigma \,\}$ the set of proper cofaces of $\tau \in K^p$, or *cofacets*, and the (weighted) *degree* of $\tau \in K^p$ with:

$$\deg_w(\tau) = \sum_{\sigma \in \mathrm{co}(\tau)} w(\sigma)$$

Note setting $w(\sigma) = 1$ for all $\sigma \in K$ recovers the integral notion of degree representing the number of cofacets a given p-simplex has. Now, since K is a simplicial complex, if the faces τ, τ' share a common cofacet $\sigma \in K^{p+1}$, this cofacet $\{\sigma\} = \mathrm{co}(\tau) \cap \mathrm{co}(\tau')$ is in fact *unique* [18]. Thus, we may use a relation $\tau \overset{\sigma}{\sim} \tau'$ to rewrite the operator from (A.7) element-wise:

$$L_p^{\mathrm{up}}(\tau, \tau') = \begin{cases} \deg_w(\tau) \cdot w^+(\tau) & \text{if } \tau = \tau' \\ s_{\tau,\tau'} \cdot w^{+/2}(\tau) \cdot w(\sigma) \cdot w^{+/2}(\tau') & \text{if } \tau \overset{\sigma}{\sim} \tau' \\ 0 & \text{otherwise} \end{cases} \tag{A.12}$$

where $s_{\tau,\tau'} = \mathrm{sgn}([\tau], \partial[\sigma]) \cdot \mathrm{sgn}([\tau], \partial[\sigma])$. Ordering the p-faces $\tau \in K^p$ along a total order and choosing an indexing function $h : K^p \to [n]$ enables explicit computation of the corresponding matrix-vector product:

$$(L_p^{\mathrm{up}}\, x)_i = \deg_w(\tau_i) \cdot w^+(\tau_i) \cdot x_i + w^{+/2}(\tau_i) \sum_{\tau_j \overset{\sigma}{\sim} \tau_i} s_{\tau_i,\tau_j} \cdot x_j \cdot w(\sigma) \cdot w^{+/2}(\tau_j) \tag{A.13}$$

Observe (A.13) can be evaluated now via a very similar two-pass algorithm as described for the graph Laplacian if the simplices of K^{p+1} can be quickly enumerated and the indexing function h can be efficiently evaluated.

Below is pseudocode outlining how to evaluate a weighted (up) Laplacian matrix-vector multiplication built from a simplicial complex K with $m = |K^{p+1}|$ and $n = |K^p|$ in essentially $O(m)$ time when $m > n$ and p is considered a small constant. Key to the runtime of the operation being essentially linear is the constant-time determination of orientation between p-faces ($s_{\tau,\tau'}$)—which can be inlined during the computation—and the use of a deterministic $O(1)$ hash table $h : K^p \to [n]$ for efficiently determining the appropriate input/output offsets to modify (i and j). Note the degree computation occurs only once.

Algorithm 1 `matvec` for weighted p up-Laplacians in $O(m(p+1)) \approx O(m)$ time ($p \geq 0$)

```
**Require:** Fixed oriented complex $K$ of size $N = |K|$
**Optional:** Weight functions $w_{p+1} : K^{p+1} \to \mathbb{R}_+$ and $w_p^l, w_p^r : K^p \to \mathbb{R}_+$
**Output:** $y = \langle L_p^{\mathrm{up}}, x \rangle = (W_p \circ \partial_{p+1} \circ W_{p+1} \circ \partial_{p+1}^T \circ W_p)x$
 1: // Precompute weighted degrees $\deg_w$
 2: Define $h : K^p \to [\, n \,]$
 3: $\deg_w \leftarrow \mathbf{0}$
 4: for $\sigma \in K^{p+1}$ do:
 5:     for $\tau \in \partial[\sigma]$ do:
 6:         $\deg_w[h(\tau)] \leftarrow \deg_w[h(\tau)] + w_p^l(\tau) \cdot w_{p+1}(\sigma) \cdot w_p^r(\tau)$
 7:
 8: function UpLaplacianMatvec($x \in \mathbb{R}^n$)
 9:     $y \leftarrow \deg_w \odot x$    (element-wise product)
10:     for $\sigma \in K^{p+1}$ do:
11:         for $\tau,\ \tau' \in \partial[\sigma] \times \partial[\sigma]$ where $\tau \neq \tau'$ do:
12:             $s_{\tau,\tau'} \leftarrow \mathrm{sgn}([\tau], \partial[\sigma]) \cdot \mathrm{sgn}([\tau'], \partial[\sigma])$
13:             $i,\ j \leftarrow\ h(\tau),\ h(\tau')$
14:             $y_i \leftarrow y_i + s_{\tau,\tau'} \cdot x_j \cdot w_p^l(\tau) \cdot w_{p+1}(\sigma) \cdot w_p^r(\tau')$
15:     return $y$
```

In general, the signs of the coefficients $\mathrm{sgn}([\tau], \partial[\sigma])$ and $\mathrm{sgn}([\tau'], \partial[\sigma])$ depend on the position of τ, τ' as summands in $\partial[\sigma]$ (2.1), which itself depends on the orientation of $[\sigma]$ (**??**). Thus, evaluation of these sign terms takes $O(p)$ time to determine for a given $\tau \in \partial[\sigma]$ with $\dim(\sigma) = p$, which if done naively via line (12) in the pseudocode A.1 increases the complexity of the algorithm. However, observe that the sign of their product is in fact invariant in the orientation of $[\sigma]$ (see Remark 3.2.1 of [18])—thus, if we fix the orientation of the simplices of K^p, the sign pattern $s_{\tau,\tau'}$ for every $\tau \overset{\sigma}{\sim} \tau'$ can be precomputed and stored ahead of time, reducing the evaluation $s_{\tau,\tau'}$ to $O(1)$ time and $O(m)$ storage. Alternatively, if the labels of the $p+1$ simplices $\sigma \in K^{p+1}$ are given an orientation induced from the total order on V, then we can remove the storage requirement entirely and simply fix the sign pattern during the computation.

A subtle but important aspect of algorithmically evaluating (A.13) is the choice of indexing function $h : K^p \to [n]$. This map is necessary to deduce the contributions of the components x_* during the operation (line (13)). While this task may seem trivial as one may use any standard associative array to generate this map, typical implementations that rely on collision-resolution schemes such as open addressing or chaining only have $O(1)$ lookup time in expectation. Moreover, empirical testing suggests that line (13) in A.1 can easily bottleneck the entire computation due to the scattered memory access such collision-resolution schemes may involve. One solution avoiding these collision resolution schemes that exploits the fact that K is fixed is to build an order-preserving *perfect minimal hash function* (PMHF) $h : K^p \to [n]$. It is known how to build PMHF's over fixed input sets of size n in $O(n)$ time and $O(n \log m)$ bits with deterministic $O(1)$ access time []. Note that this process happens only once for a fixed simplicial complex K: once h has been constructed, it is fixed for every `matvec` operation.

Directional Transform

The canonical interpretation of the information displayed by a persistence diagram is that is summarizes the persistence of the sublevel sets of filtered space. Given a filtration pair $(\, K, f\,)$ where K is a finite simplicial complex and $f : K \to \mathbb{R}$ is a real-valued function, the sublevel sets $|K|_i = f^{-1}(-\infty, i]$ deformation retract to... If K is embedded in $\mathbb{R}^d$, then geometrically f takes on the interpretation of a 'height' function whose range yields the 'height' of every simplex in K.

Let $X \subset \mathbb{R}^d$ denote a data set which can be written as a finite simplicial complex K whose simplices are PL-embedded in $\mathbb{R}^d$. Given this setting, define the *directional transform* (DT) of K as follows:

$$\mathrm{DT}(K) : S^{d-1} \to K \times C(K, \mathbb{R})$$
$$v \mapsto (K_\bullet, f_v)$$

where we write $(K_\bullet, f)$ to indicate the filtration on K induced by f_v for all $\alpha \in \mathbb{R}$, i.e.:

$$K_\bullet = K(v)_\alpha = \{\, x \in X \mid \langle x, v \rangle \leq \alpha \,\} \tag{A.14}$$

Conceptually, we think of DT as an S^{d-1}-parameterized family of filtrations.

The Persistent Homology Transform (PHT) is a shape statistic that establishes a fundamental connection between the topological information summarized by K's PH groups and the geometry of its associated embedding. Given a complex K built from X, it is defined as:

$$\begin{aligned} \mathrm{PHT}(K) : S^{d-1} &\to \mathcal{D}^d \\ v &\mapsto \big(\mathrm{dgm}_0(K, v), \mathrm{dgm}_1(K, v), \ldots, \mathrm{dgm}_{d-1}(K, v) \big) \end{aligned} \tag{A.15}$$

where $\mathcal{D}$ denotes the space of p-dimensional persistence diagrams, for all $p = 0, \ldots, d-1$ and S^{d-1} the unit $d-1$ sphere. The stability of persistence diagrams ensures that the map $v \mapsto \mathrm{dgm}_p(K, v)$ is Lipschitz with respect to the bottleneck distance metric $d_B(\cdot, \cdot)$ whenever K is a finite simplicial complex. Thus, the PHT may be thought of as an element in $C(S^{d-1}, \mathcal{D}^d)$: .

The primary result of [] is that the PHT is injective on the space of subsets of R^d that can be written as finite simplicial complexes[9], which we denote as $\mathcal{K}_d$. Equivalently, $\mathcal{K}_d$ decomposes space of all pairs (K, f) under the equivalence $(K, f) \sim (K, f')$ when $f(K) = f'(K)$.

A.2 Complexity of Persistence & Related work

We briefly recount the main complexity results of the persistence computation. With a few key exceptions, the majority of persistent homology implementations and extensions is based on the *reduction algorithm* introduced by Edelsbrunner and Zomorodian [17]. This algorithm factorizes the filtered boundary into a decomposition $R = \partial V$, where V is full rank upper-triangular and R is said to be in reduced form: if its i-th and j-th columns are nonzero, then $\mathrm{low}_R(i) \neq \mathrm{low}_R(j)$, where $\mathrm{low}_R(i)$ denotes the row index of the lowest non-zero in column i. We refer to [17, 2, 14] for details.

Given a filtration (K, f) of size $m = |K|$ with filter $f : K \to [m]$, the reduction algorithm in form given in [17] computes $\mathrm{dgm}_p(K; \mathbb{Z}/2) = \{\, (\tau_1, \sigma_1), (\tau_2, \sigma_2), \ldots, (\tau_k, \sigma_k) \,\}$ runs in time proportional to the sum of the squared (index) persistences $\sum_{i=1}^{k} (f(\sigma_i) - f(\tau_i))^2$. As k is at most $m/2$, this implies a $O(m^3)$ upper bound on the complexity of the general persistence computation, which incidentally Morozov showed was a tight $\Theta(m^3)$ under the assumption that each column reduction takes $O(m)$ time. By exploiting the matrix-multiplication results, a similar result can be shown to reduce to $O(m^\omega)$, where ω is the matrix-multiplication constant, which is ≈ 2.37 as of this time of writing. It worth remarking that the complexity statements above are all given in terms of the number of *simplices* m: if $n = |K^0|$ is the size of the vertex set, the above implies a worst-case bound of $O(n^{\omega(p+2)})$ on the general persistence computation. For example, if we use non-Strassen-based matrix multiplication ($\omega = 3$) and we are concerned with $p = 1$ homology computation, the complexity of the reduction algorithm scales $O(n^9)$ in the number of vertices of the complex, which is essentially intractable for most real world application settings.

Despite the seemingly immense intractability of the persistence computation, decades of advancements have been made in reducing the complexity or achieving approximate results in reasonable time and space complexities. The complexity of the reduction algorithm is complicated by the fact that it depends heavily on the structure of the associated filtration K, the homology dimension p, the field of coefficients $\mathbb{F}$, and the assumptions about the space K manifests from. In [], Sheehy presented an algorithm for producing a sparsified version $(\tilde{K}, \tilde{f})$ of a given Vietoris-Rips filtration (K, f) constructed from an n-point metric space (X, d_X) whose total number of p-simplices is bounded above by $n \cdot (\epsilon^{-1})^{O(pd)}$, where d is the doubling dimension of X. It was shown that $\mathrm{dgm}_p(\tilde{K})$ is guaranteed to be a multiplicative c-approximation to the $\mathrm{dgm}_p(K)$, where $c = (1 - 2\epsilon)^{-1}$ and $\epsilon \leq 1/3$ is a positive approximation parameter. When $p = 0$ and the filtration function $f : K \to \mathbb{R}$ is PL, the reduction algorithm can be bypassed entirely in favor of simple $O(n \log n + \alpha(n)m) \approx O(m)$ algorithm (see Algorithm 5 in [14]), where $n = |K^0|$ and $m = |K^1|$ and $\alpha(n)$ is the extremely slow-growing inverse Ackermann function. Moreover, the $d-1$ persistence pairs can be

[9] Implicit in the injectivity statement of the PHT is that, given a subset $X \subset \mathbb{R}^d$ which may be written as finite simplicial complex K, the restriction $f : X \to \mathbb{R}$ to any simplex in K must is linear.

computed in $O(n\alpha(n))$ time algorithm for filtrations of simplicial d-manifolds essentially reducing the problem to computing persistence on a dual graph [14]. For clique complexes, the apparent pairs optimization—which preemptively removes zero-persistence pairs from the computation prior to the reduction—has been empirically observed to reduce the number of columns needing reduced for clique complexes by $\approx 98 - 99\%$ [2]. Numerous other optimizations, including e.g. the *clearing optimization*, the use of *cohomology*, the *implicit reduction* technique, have further reduced both the non-asymptotic constant factors of the reduction algorithm significantly, see [2] and references therein for a full overview.

Despite the dramatic reductions in time and space needed for the persistence algorithm to complete, to the author knowledge relatively little has been done in improving the complexity and effective runtime of the reduction in parameterized settings. Although both of these algorithms have shown significant constant-factor reductions in the (re)-reduction of the associated sparse matrices, all of the techniques require $O(m^2)$ storage to execute as the R and V matrices must be maintained throughout the computation. Moreover, all three of the above methods intrinsically work within the reduction framework, wherein simulating persistence in dynamic contexts effectively reduces to the combinatorial problem of maintaining a valid $R = \partial V$ decomposition.

As noted in [14], the reduction algorithm is essentially a variant of Gaussian elimination. Indeed, the persistence of a given filtration can be computed by the PLU factorization of a matrix. The explicit decompositional approach of factorizing a large matrix into constitutive parts is known historically in numerical linear algebra as a *direct method*—methods would yield the exact solution within a finite number of steps. In contrast, iterative methods start with approximate solution and progressively update the solution up to arbitrary accuracy. The iterative methods well-known to the numerical linear algebra community, such as Krylov methods, are typically often attractive not only due to the reduction in computational work over direct approaches but also of the limited amount of memory that is required. Despite the success of iterative methods in efficiently solving linear systems manifesting from diagonally dominant sparse matrices is [], such advancements have not yet been extended to the persistence setting.

Output sensitive multiplicity and Betti

We record this fact formally with two corollaries. Let $\mathrm{R}_p(k)$ denotes the complexity of computing the rank of square $k \times k$ matrix with at most $O((p+1)k)$ non-zero $\mathbb{F}$ entries. Then we have:

Corollary 4. *Given a filtration $K_\bullet$ of size $N = |K_\bullet|$ and indices $(\, i, j\,) \in \Delta_+^N$, computing $\beta_p^{i,j}$ using expression* (2.6) *requires $O\big(\max\{\mathrm{R}_p(n_i), \mathrm{R}_{p+1}(m_j)\}\big)$ time, where $n_i = |K_i^p|$ and $m_j = |K_j^{p+1}|$.*

Observe the relation $\partial_{p+1}^{i+1,j} \subseteq \partial_{p+1}^{1,j}$ implies the dominant cost of computing (2.6) lies in computing either $\mathrm{rank}(\partial_p^{1,i})$ or $\mathrm{rank}(\partial_{p+1}^{1,j})$, which depends on the relative sizes of $|K^p|$ and $|K^{p+1}|$. In contrast, μ_p^R is localized to the pair (K_i, K_l) and depends only on the $(p+1)$-simplices in the interval $[i, l]$, yielding the following corollary.

Corollary 5. *Given a filtration $K_\bullet$ of size $N = |K_\bullet|$ and a rectangle $R = [i, j] \times [k, l]$ with indices $0 \leq i < j \leq k < l \leq N$, computing μ_p^R using expression* (2.7) *requires $O(\mathrm{R}_{p+1}(m_{il}))$ time $m_{il} = |K_l^{p+1}| - |K_i^{p+1}|$.*

A.3 Laplacian Interpretation

In what follows we make a connection between boundary matrices and the graph Laplacian to illustrate how the Laplacian captures the "connectivity" aspects of the underlying simplicial complex.

Example A.1 (Adapted from [25])**.** Suppose the ordered vertices of G are labeled from 1 to n such that, given any subset $X \subseteq V$, we may define column vector $x = (\, x_i \,)$ whose components $x_i = 1$ indicate $i \in X$ and $x_i = 0$ otherwise. Then, given $X \subseteq V$ and its complement set $X' = V \setminus X$, we have:

$$\begin{aligned}
(Lx)_i > 0 &\iff i \in X \text{ and } |c_i(X)| = (Lx)_i \\
(Lx)_i < 0 &\iff i \in X' \text{ and } |c_i(X')| = |(Lx)_i| \\
(Lx)_i = 0 &\iff i \in X \cup X' \text{ and } c_i(X) = \emptyset
\end{aligned}$$

where $c_v(X) = \{(v, w) \in E \mid v \in X \text{ and } w \in V \setminus X\}$ denotes the *cutset* of X restricted to v, i.e. the set of edges having as one endpoint $v \in X$ and another endpoint outside of X.

In other words, example A.1 demonstrates that L captures exactly how X is connected to the rest of G. Notice that if $X = V$, then $Lx = 0$ and thus 0 must be an eigenvalue of L with an eigenvector pair $\mathbf{1}$. Like the adjacency matrix, the interpretation of the matrix-vector product has a natural extension to powers of L, wherein just as entries in A^k model paths, entries in L^k are seen to model boundaries [25].

Parameterizing Settings

We include a few examples of potential application areas of work. Namely, we show a few promising examples of "parameterized settings" that may naturally benefit from our efforts here.

Dynamic Metric Spaces: Consider an $\mathbb{R}$-parameterized metric space $\delta_X = (X, d_X(\cdot))$ where X is a finite set and $d_X(\cdot) : \mathbb{R} \times X \times X \to \mathbb{R}_+$, satisfying:

1. For every $t \in \mathbb{R}, \delta_X(t) = (X, d_X(t))$ is a pseudo-metric space[10]
2. For fixed $x, x' \in X$, $d_X(\cdot)(x, x') : \mathbb{R} \to \mathbb{R}_+$ is continuous.

When the parameter $t \in \mathbb{R}$ is interpreted as *time*, the above yields a natural characterization of a "time-varying" metric space. More generally, we refer to an $\mathbb{R}^h$-parameterized metric space as *dynamic metric space*(DMS). Such space have been studied more in-depth [] and have been shown...

A.4 Proofs

Proof of rank equivalence

In general, it is not true that $\text{rank}(A) = \text{rank}(\text{sgn}(A))$. However, it is true that $\text{rank}(\partial_p) = \text{rank}(\text{sgn}(\partial_p))$.

Proof of Lemma 1

Proof. The Pairing Uniqueness Lemma [14] asserts that if $R = \partial V$ is a decomposition of the total $m \times m$ boundary matrix ∂, then for any $1 \leq i < j \leq m$ we have $\text{low}_R[j] = i$ if and only if $r_\partial(i, j) = 1$. As a result, for $1 \leq i < j \leq m$, we have:

$$\text{low}_R[j] = i \iff r_R(i, j) \neq 0 \iff r_\partial(i, j) \neq 0 \tag{A.16}$$

Extending this result to equation (2.5) can be seen by observing that in the decomposition, $R = \partial V$, the matrix V is full-rank and obtained from the identity matrix I via a sequence of rank-preserving (elementary) left-to-right column additions. □

Proof of Proposition 1

Proof. We first need to show that $\beta_p^{i,j}$ can be expressed as a sum of rank functions. Note that by the rank-nullity theorem, so we may rewrite (3.1) as:

$$\beta_p^{i,j} = \dim\left(C_p(K_i)\right) - \dim\left(B_{p-1}(K_i)\right) - \dim\left(Z_p(K_i) \cap B_p(K_j)\right)$$

The dimensions of groups $C_p(K_i)$ and $B_p(K_i)$ are given directly by the ranks of diagonal and boundary matrices, yielding:

$$\beta_p^{i,j} = \text{rank}(I_p^{1,i}) - \text{rank}(\partial_p^{1,i}) - \dim\left(Z_p(K_i) \cap B_p(K_j)\right)$$

To express the intersection term, note that we need to find a way to express the number of p-cycles born at or before index i that became boundaries before index j. Observe that the non-zero columns of R_{p+1} with index at most j span $B_p(K_j)$, i.e $\{\, \text{col}_{R_{p+1}[k]} \neq 0 \mid k \in [j] \,\} \in \text{Im}(\partial_{p+1}^{1,j})$. Now, since the low entries of the non-zero columns of R_{p+1} are unique, we have:

$$\dim(Z_p(K_i) \cap B_p(K_i)) = |\Gamma_p^{i,j}| \tag{A.17}$$

[10] This is required so that if one can distinguish the two distinct points $x, x' \in X$ incase $d_X(t)(x, x') = 0$ at some $t \in \mathbb{R}$.

where $\Gamma_p^{i,j} = \{\, \mathrm{col}_{R_{p+1}[k]} \neq 0 \mid k \in [j], 1 \leq \mathrm{low}_{R_{p+1}}[k] \leq i \,\}$. Consider the complementary matrix $\bar{\Gamma}_p^{i,j}$, given by the non-zero columns of R_{p+1} with index at most j that are not in $\Gamma_p^{i,j}$, i.e. the columns satisfying $\mathrm{low}_{R_{p+1}}[k] > i$. Combining rank-nullity with the observation above, we have:

$$|\bar{\Gamma}_p^{i,j}| = \dim(B_p(K_j)) - |\Gamma_p^{i,j}| = \mathrm{rank}(R_{p+1}^{i+1,j}) \tag{A.18}$$

Combining equations (A.17) and (A.18) yields:

$$\dim(Z_p(K_i) \cap B_p(K_j)) = |\Gamma_p^{i,j}| = \dim(B_p(K_j)) - |\bar{\Gamma}_p^{i,j}| = \mathrm{rank}(R_{p+1}^{1,j}) - \mathrm{rank}(R_{p+1}^{i+1,j}) \tag{A.19}$$

Observing the final matrices in (A.19) are *lower-left* submatrices of R_{p+1}, the final expression (2.6) follows by applying Lemma 1 repeatedly. □

Proof of boundary matrix properties

Proof. First, consider property (1). For any $t \in T$, applying the boundary operator ∂_p to $K_t = \mathrm{Rips}_\epsilon(\delta_{\mathcal{X}}(t))$ with non-zero entries satisfying (**??**) by definition yields a matrix ∂_p satisfying $\mathrm{rank}(\partial_p) = \dim(\mathrm{B}_{p-1}(K_t))$. In contrast, operators of the from (3.4) always produce p-boundary matrices of Δ_n; however, notice that the only entries which are non-zero are precisely those whose simplices σ that satisfy $\mathrm{diam}(\sigma) < \epsilon$. Thus, $\mathrm{rank}(\partial_p^t) = \dim(\mathrm{B}_{p-1}(K_t))$ for all $t \in T$. < (show proof of (2))> Property (3) follows from the construction of ∂_p and from the inequality $\|A\|_2 \leq \sqrt{m}\|A\|_1$ for an $n \times m$ matrix A, as $\|\partial_p^t\|_1 \leq (p+1)\,\epsilon$ for all $t \in T$. □

A.5 Proofs of basic properties

Proof. The above result immediately follows by applying the fact that $\lim_{\tau \to 0^+} \|\Phi_\tau(X)\|_* = \mathrm{rank}(X)$ to each of the constitutive terms of $\hat{\mu}_{p,\tau}^R$ and $\hat{\beta}_{p,\tau}^{i,j}$. □

B Boundary matrix factorization

Definition 2 (Boundary matrix decomposition)**.** Given a filtration $K_\bullet$ with m simplices, let ∂ denote its $m \times m$ filtered boundary matrix. We call the factorization $R = \partial V$ the *boundary matrix decomposition* of ∂ if:

I1. V is full-rank upper-triangular

I2. R satisfies $\mathrm{low}_R[i] \neq \mathrm{low}_R[j]$ iff its i-th and j-th columns are nonzero

where $\mathrm{low}_R(i)$ denotes the row index of lowest non-zero entry of column i in R or null if it doesn't exist. Any matrix R satisfying property (I2) is said to be *reduced*; that is, no two columns share the same low-row indices.

C Laplacian facts

In general, the spectrum of the graph Laplacian L is unbounded, [] and instead many prefer to work within the "normalized" setting where eigenvalues are bounded. The *normalized Laplacian* $\mathcal{L}$ of a graph G is typically given as:

$$\mathcal{L}(G) = D^{-1/2} L D^{-1/2} \tag{C.1}$$

with the convention that $D^{-1}(v_i, v_i) = 0$ for $\deg(v_i) = 0$. The variational characterization of eigenvalues in terms of the Rayleigh quotient of $\mathcal{L}$ convey a particular form. Specifically, for any real-valued function $f : V \to \mathbb{R}$ on G, when viewed as a column vector, $\mathcal{L}$ satisfies:

$$\frac{\langle f, \mathcal{L} f \rangle}{\langle f, f \rangle} = \frac{\sum\limits_{i \sim j} (g(v_i) - g(v_j))^2}{\sum\limits_{i} g(v_i)^2 \cdot \deg(v_i)} \tag{C.2}$$

where $f = D^{1/2}g$ and $\langle f, g\rangle$ denotes the standard inner product in $\mathbb{R}^n$. Equation (C.2) may be used to show that the spectrum $\Lambda(\mathcal{L})$ is bounded in the interval $[0, 2]$. In particular, it is known that:

$$\lambda_i \leq \sup_f \frac{\langle f, \mathcal{L} f\rangle}{\langle f, f\rangle} \leq 2 \tag{C.3}$$

Recall that, when G is connected, 0 is an eigenvalue of both L and $\mathcal{L}(G)$, with multiplicity $\mathrm{cc}(G)$. Moreover, if G is the union of disjoint graphs $G_1, G_2, \ldots, G_k$, then it has as its spectrum the union of the spectra $\Lambda(G_1), \Lambda(G_2), \ldots, \Lambda(G_k)$. Certain parts of the spectrum of $\mathcal{L}$ can be deduced explicitly for very structured types of G, such as complete graphs, complete bipartite graphs, star graphs, path graphs, and cycle graphs, and n-cubes. For a list of additional properties the graph and normalized Laplacians satisfy, including bounds on eigenvalues, relation to random walks and rapidly-mixing Markov chains, identities tied to isoperimetric properties of graphs, and explicit connections to spectral Riemannian geometry, see [11] and references within.

D Parameterized setting & Perturbation theory

If f is a real-valued filter function that variously smoothly in $\mathcal{H}$, one would expect the spectra of the constitutive terms in β_p^* and μ_p^* to also vary smoothly as functions of $\mathcal{H}$. Indeed, since Laplacian matrices are normal matrices, we expect their spectra to be quite stable under perturbations [].

Small condition numbers often improve the convergence of iterative solvers and improve stability of spectrum with respect to perturbations in the entries of the matrix. $\kappa(M^{-1}A)$

$$M^{-1}Ax = M^{-1}b$$

where M is symmetric positive definite.

$$\min_{x \perp \mathbf{1}} \frac{1}{2} x^T(L + \epsilon I_n)x - b^T x \tag{D.1}$$

Since this nonsingular, positive definite, strictly diagonally dominant matrix, thus we may apply the famous Conjugate Gradient (CG) algorithm to solve such a system. It's well known that CG converges to the solution of $Ax = b$ in exactly $O(n)$ iterations (and often much earlier), of which each iteration requires one $O(m)$ matrix-vector product, implying a runtime of $O(mn^2)$ (compare with...). Moreover, and since this is a Laplacian matrix, the wealth of tools developed for said matrices may also be used. In particular, [] showed that *low-stretch spanning trees* act as good preconditioners to accelerate Laplacian solvers, wherein it's been shown that the preconditioned Conjugate Gradient (PCG) requires as most $O(\sqrt{m}\log n)$ iterations, each of which requires one matrix-vector product using L_G and in $O(m^{1/3} \log n \ln 1/\epsilon)$ iterations. This was later improved by, who showed that one can solve Laplacian systems effectively in $O(m \log^{O(1)} n)$ time, giving a bound of $O(rm \log^{O(1)} n)$ time to obtain....

Of course, if one wants to compute either of the counting invariants in... exactly for $p = 0$, of course, the fastest algorithm is to reduce the problem to the well-known elder-rule problem, which takes $O(m \log m + m\alpha(n))$ time for a general filtration. It is unlikely that we may beat this bound, either in theory or in practice, for $p = 0$. However, the fastest known algorithm for computing the full persistence diagram for $p \geq 1$ is $O()$, which is quite a jump in complexity; there is no generalization of disjoint-set algorithm for the case where $p \geq 1$. Moreover, these direct methods tend to be memory bound operations, pushing researchers who want to compute these diagrams in practice to focus on ways of reducing the memory usage, such as using $\mathbb{Z}_2$ field coefficients. In contrast, the means by which we compute these invariants scales quite well with larger p, it produces a stronger invariant, and is far more reaching to other areas of mathematics.