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 [19] (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 [14]. Whether for shape recognition [10], dimensionality reduction [37], or time series analysis [33], 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 [7], and template functions [34]. Tackling the issue of injectivity, Turner et al. [40] 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, scalability issues due to high algorithmic complexity remain. Indeed, this issue is compounded in the parameterized setting, where adaptations of the persistence computation has proven non-trivial [35].

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 [9, 11], 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 also *matrix-free*, circumventing the reduction algorithm from [18] entirely. Additionally, the relaxation is computable exactly in linear space and quadratic time, requires no complicated data structures or maintenance procedures to implement,

and can be iteratively $(1 \pm \epsilon)$ approximated in effectively linear time in practice for large enough $\epsilon > 0$.

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 (section 4). Moreover, both relaxations admit iterative $(1 \pm \epsilon)$-approximation schemes**??**, recovering both invariants ϵ is 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 [25].

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:

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

4. Choose a Laplacian operator $\mathcal{L}_p^{a,b}$ to act on the relative p-cochains of (K_b, K_b) and a spectral function $\phi(\cdot, \tau) : \mathbb{R} \to \mathbb{R}_+$ which converges to sgn_+ as $\tau \to 0$. The choice of $\mathcal{L}_p^*$ (e.g. Kirchoff, random walk) determines the geometric/topological information to extract from (K, f_α) and 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 $\mathcal{R}$:

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

   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 with their Laplacian spectra; examples include topology-guided image denoising [36], shape classification under metric invariants [10], bifurcation detection in dynamical systems [34], 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 [31].

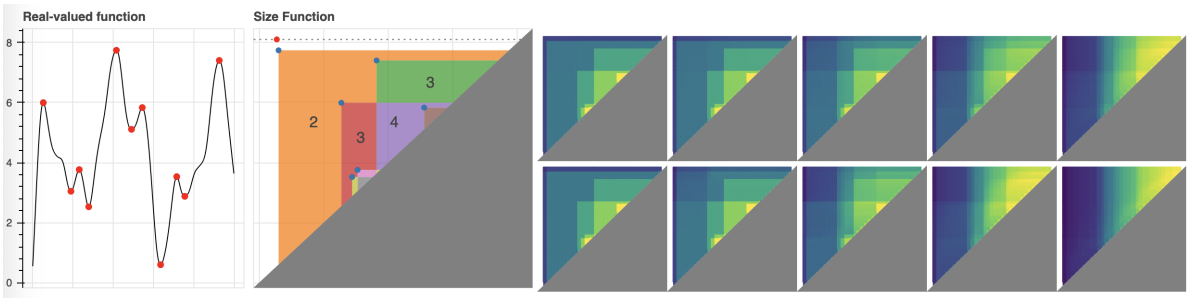


Figure 1: (left) A function $f : \mathbb{R} \to \mathbb{R}$ and its corresponding persistence diagram obtained by filtering f by its sublevel sets. (right) two spectral interpolations

2 Notation & Background

A *simplicial complex* $K \subseteq \mathcal{P}(V)$ over a finite set $V = \{v_1, v_2, \ldots, v_n\}$ 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 $[\sigma] = (-1)^{|\pi|} \left[v_{\pi(1)}, v_{\pi(2)}, \ldots, v_{\pi(p+1)}\right]$, where π is a permutation on $[\,p+1\,] = \{\,1, 2, \ldots, p+1\,\}$ 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:

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

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 [43]: 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 $\operatorname{dgm}_p(K, f)$ 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 [25], 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 $\mathcal{F} : \mathbb{R}^d \to \mathbf{Simp}$ satisfying $\mathcal{F}(a, b) : \mathcal{F}_a \to \mathcal{F}_b$ an inclusion for all $a \leq b$ []. By fixing $K \in \mathbf{Simp}$ 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 [43], which shows 1-parameter persistence modules decompose in an *essentially unique* way. Computationally, the corresponding Pairing Uniqueness Lemma [15] 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:

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

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:*

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

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 [16], though the latter was also noted in passing by Edelsbrunner [19]. 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 [4]:

Corollary 1 (Bauer et al. [4])**.** *Any persistence algorithm which preserves the ranks of the submatrices $\partial^{i,j}(K, f)$ for all $i, j \in [N]$ satisfying $1 \leq i < j \leq N$ 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 ([12, 16])**.** *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:*

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

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

Though (2.7) was pointed out by Cohen-Steiner et al. in [15] and exploited computationally by Chen & Kerber in [12], to the authors knowledge the only explicit derivation and proof of (2.6) is given by Dey & Wang [16] (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.

Two important properties of the expressions from Corollary 2 are: (1) they are comprised strictly of *rank* computations, and (2) all terms involve *unfactored* boundary matrices. Coupled with measure-theoretic perspectives on persistence [11], the former suggests variational perspectives of the rank function might yield interesting spectral relaxations of (2.6) and (2.7) useful for e.g. optimization purposes. Combined with Corollary 1, the latter property suggests a path to compute persistence information *without* matrix reduction. Moreover, combining these observations suggest advances made in other areas of applied mathematics may be readily exploited, such as the rich theory of matrix functions [5], the tools developed as part of "The Laplacian Paradigm" [39], or the recent connections between rank and trace estimation [41]. The rest of the paper is dedicated to exploring these connections and their implications.

3 Spectral relaxation and its implications

Prior to introducing our proposed relaxation, 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 [5, 22] or persistence-based [29, 43] 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. [15]), a decomposition $R = \partial V$ of the boundary matrix ∂ must be permuted and modified frequently to maintain a simplexwise order respecting f_α. In contrast, one benefit of the rank function is its permutation invariance: for any $X \in \mathbb{R}^{n\times n}$ and permutation P we have:

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

This suggests persistent rank computations like those from Corollary 2 need not maintain this ordering—as long as the constitutive boundary matrices the same non-zero pattern as their filtered counterparts, their ranks will be identical. In what follows, we demonstrate how exploiting this permutation invariance significantly simplifies the practical use of (2.6) and (2.7) in *parameterized* settings.

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 $s_{kl} = \operatorname{sgn}([\sigma_k], \partial[\sigma_l])$ is the sign of the oriented face $[\sigma_k]$ in $\partial[\sigma_l]$. Observe that $\partial_p(\alpha)$ may be decoupled (3.2) 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, $\operatorname{rank}(\partial_p(\alpha)) = \operatorname{rank}(\partial_p(K_\preceq))$ 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:

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

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).

The observations above collectively motivate our first relaxation. Without loss in generality, assume the orientation of the simplices $(K, \preceq)$ is induced by 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 $\beta^{a,b}_p : \mathcal{A} \times K \to \mathbb{N}$ and $\mu^R_p : \mathcal{A} \times K \to \mathbb{N}$ defined by:*

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

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

yield the correct quantities $\mu^R_p(K, f_\alpha) = \operatorname{card}\Big(\operatorname{dgm}_p(f_\alpha)\big|_R\Big)$ and $\beta^{a,b}_p = \dim(H^{a,b}_p(K, f_\alpha))$ 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.

Corollary 3. *Given a boundary operator $\hat{\partial}_p(\alpha) \in \mathbb{R}^{n\times n}$ constructed at time $\alpha \in \mathbb{R}$ from a parameterized family of filtrations (K, f_α) and a fixed linear extension $\preceq$, the time complexity of constructing $\hat{\partial}_p(\alpha')$ for any other $\alpha' \neq \alpha$ is $O(\max(|K^p|, |K^{p+1}|))$, assuming the evaluation of $f_\alpha(\tau)$ is $O(1)$ for every simplex $\tau \in K$.*

3.2 Parameterized Laplacians

For generality's sake, it is important to make the class of expressions for β^*_p and μ^*_p as large as possible. Since we are only concerned with homology over $\mathbb{R}$, we may exploit another identity of the rank function which is only applicable to zero characteristic fields:

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

In the context of boundary operators, note that $\partial_1\partial_1^T$ is the well known *graph Laplacian* [13], indicating we may express $\beta^*_0(\alpha)$ and $\mu^*_0(\alpha)$ using the ranks of Laplacian[2] matrices.

Following the seminal results from Horuk and Jost [24], there are three natural ways to define (p-) Laplacian operators over a fixed simplicial complex K: the *up*-Laplacian $L^{\text{up}}_p(K)$, the *down*-Laplacian $L^{\text{dn}}_p(K)$, and their sum, which we refer to as the *combinatorial* Laplacian $\Delta_p(K)$:

$$\Delta_p = \underbrace{\partial_{p+1} \circ \partial^T_{p+1}}_{L^{\text{up}}_p} + \underbrace{\partial^T_p \circ \partial_p}_{L^{\text{dn}}_p} \tag{3.7}$$

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

All three operators Δ_p, L_p^{up}, and L_p^{dn} are symmetric, positive semi-definite, and compact [29]—moreover, the *non-zero* multisets $\Lambda(L_p^{\text{up}})$ and $\Lambda(L_{p+1}^{\text{dn}})$ are equivalent, implying they must have identical ranks (see Theorem 2.2 and 3.1 of [24]). Thus, for rank computations, it suffices to consider only one of them.

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 $\mathcal{L}_p \triangleq L_p^{\text{up}}$ at index (a, b) as follows:

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

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:

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

where the second last equality uses the identity $\operatorname{rank}(X) = \operatorname{rank}(X^T X)$. 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 the action of sending a subset $S \subseteq K$ of p-simplices to 0 as a restriction of K to a sub-complex $L = K \setminus S$, which suggests e.g. (3.1) and (3.8) could be alternatively defined using the inclusion maps $L \hookrightarrow K$ between *simplicial pairs* (L, K), as in [29]. We prefer the use of index notation (a, b) here as it is simpler notationally to generalize to arbitrary rectilinear subsets $\mathcal{R} \subset \Delta_+$.

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 function. In contrast, due to the rank function, the quantities from Proposition 1 are discontinuous functions by definition, as they are integer-valued. To understand how these discontinuities arise, we consider the spectral characterization of the rank function:

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

In the above, $\{\sigma_i\}_{i=1}^n$ are the singular values $\Sigma = \operatorname{diag}(\{\sigma_i\}_{i=1}^n)$ from the singular value decomposition (SVD) $X = U\Sigma V^T$ of $X \in \mathbb{R}^{n \times m}$, and $\operatorname{sgn}_+ : \mathbb{R} \to \{0, 1\}$ is the one-sided sign function. As the singular values vary continuously under perturbations in X [5], 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. [28]. 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 [6]:

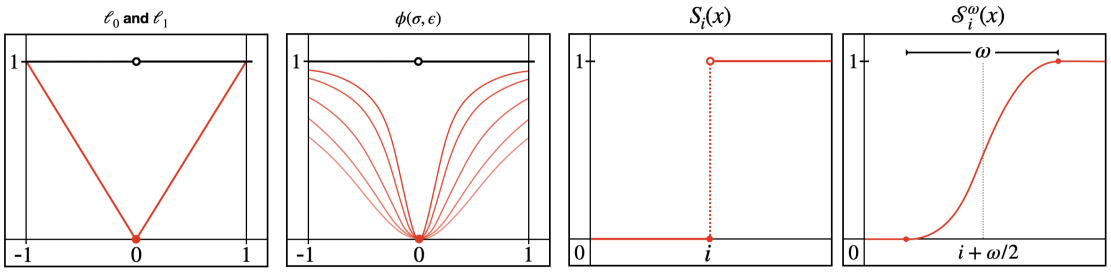


Figure 2: From left to right—the ℓ_1 norm (red) forms a convex envelope over the ℓ_0 (black) pseudo-norm on the interval $[-1, 1]$; $\phi(\cdot, \tau)$ from (3.14) at various values of $\tau > 0$ (red) and at $\tau = 0$ (black); the step function $S_i(x)$ from (3.4); the smoothstep relaxation $\mathcal{S}_i^{\omega}$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

Observe Definition (3.12) is closely related to that of a *matrix function* $f(A) \triangleq U f(\Lambda) U^T$ defined over square matrices [5]. Due to the restrictions on ϕ (3.11), the operator Φ_τ exhibits a variety of attractive properties related to rank-approximation, monotonicity, and differentiability.

Proposition 2 (Bi et al. [6])**.** *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 $\sum_{i=1}^{n} \phi(\sigma_i, \tau)$*
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 [6]. $\|\Phi_\tau(X)\|_*$ is differentiable on the positive semi-definite cone $\mathbb{S}_+^n$.

Interpretation #1: In many applications, it is common to regularize an ill-posed objective function to encourage simpler solutions or to prevent overfitting. For example, the classical least-squares approach to solving the linear system $Ax = b$ is often augmented with the *Tikhonov regularization* (TR) for some $\tau > 0$:

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

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):

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

[3] Here, "semismooth" refers to the existence certain directional derivatives in the limit as $\tau \to 0^+$, see [5, 6].
[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 [17])

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

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

The relaxation level τ may be thought of as a regularization term 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:

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

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 a spectral rank relaxation using (3.18) as a *geometrically informative RI approximation*.

4 Computational Implications

4.1 Exact computation

As evidenced by Section 3, computing $\hat{\mu}_p^*$ and $\hat{\beta}_p^*$ may be reduced to computing eigenvalues of p-Laplacians. To do this efficiently, we employ the *Lanczos method* [26], 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 triplet:

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

[5] See Theorem 2 of [42] 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.

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 that only a matrix-vector product operator $v \mapsto Av$ is required for execution—A need not necessarily be stored in memory explicitly. This aspect alone suggests Lanczos may be a viable low-memory option for computing eigenvalues. Indeed, due to its *three-term recurrence* [38], the Lanczos method requires just three $O(n)$-sized vectors and a few $O(n)$ vector operations to obtain T—neither K nor Q need be formed explicitly.

Lemma 2 ([32])**.** *Given a symmetric rank-r matrix $A \in \mathbb{R}^{n\times n}$ whose matrix-vector operator $x \mapsto Ax$ 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 result is well established for the graph Laplacian, it is not immediately clear whether a similar guarantee 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 $x \mapsto L_p x$ in $O(m(p+1))$ time and $O(\max(c, m))$ space.*

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—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 multiplicity function $\mu_p^R(K)$ derived from a simplicial complex K with $n_{ad} = |K_d^p| - |K_a^p|$ and $m_{ad} = |K_d^{p+1}| - |K_a^{p+1}|$, can be computed in exact arithmetic in the following time and space complexities:*

$$\mu_p^R(K) \overset{\text{time}}{=} O(n_{ad} \cdot m_{ad}), \quad \mu_p^R(K) \overset{\text{space}}{=} O(\max(n_{ad}, m_{ad}))$$

In particular, when $R = [-\infty, *]\times[*, +\infty]$, $\mu_p^R(K)$ has time and space complexities of $O(nm)$ and $O(\max(n, m))$, respectively, where $n = |K^p|$ and $m = |K^{p+1}|$.

It's worth noting that the standard reduction-family of algorithms computes the p-th persistent homology of a filtration K of dimension $p + 1$ and of size $N = |K| \sim O(|K^{p+1}|)$ in $O(N^3)$ time and $O(N^2)$ space, respectively—these bounds are actually tight $\Theta(N^3)$. Interestingly, Chen and Kerber [12] 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. Since both $|K^p|$ and $|K^{p+1}|$ are trivially bounded above by $O(N)$, by Proposition 3, we may recover the same $O(N^3)$ time complexity of the reduction algorithm using only rank computations, and we improve the space complexity by a factor of N, though at the cost of not having immediate access to cycle representatives.

Remark 3. A similar result for computing *non-persistent* Betti numbers of simplicial complexes over finite fields was given by Edelsbrunner and Parsa in [], who showed that the complexity of computing the Betti numbers of a 2-dimensional simplicial complex K with n vertices reduces is $\Omega(r(n, m))$, where $r(n, m)$ is the complexity of computing the rank of an n-by-n binary matrix with m non-zero entries.

[7]For example, the Lanczos method has expected time complexity $O(nzr)$ for sparse matrices containing an average of z nonzeros per row [22]

4.2 Randomized (η, ϵ)-approximation via trace estimation

As in [32], 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—among other difficulties, these errors affect both the convergence rate and termination conditions of the algorithm [32], prohibiting its use practically. Though many improvements have been proposed (e.g. selective re-orthogonalization, implicit restarting), it is well-known that the Lanczos method is not efficient at obtaining accurate eigenvalue approximations on the interior of the spectrum.

Fortunately, accurately estimating eigenvalues is not necessary for accurately estimating the rank. Unlike eigenvalue estimation, it has been shown that using Lanczos in finite precision arithmetic is stable for matrix function[8] approximation $x \mapsto f(A)x$:

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

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 above results underpin the *stochastic Lanczos quadrature* (SLQ) method employed by Ubaru et. al [41], who propose the use of SLQ method for estimating the trace of a matrix function by means of a Girard-Hutchinson (GH) estimator:

$$\mathrm{tr}(f(A)) = \sum_{i=1}^{n} e_i^T f(A) e_i \approx \frac{n}{n_v} \sum_{j=1}^{n_v} e_1^T f(T_m^{(j)}) e_1 = \frac{n}{n_v} \sum_{j=1}^{n_v} \left(\sum_{i=1}^{m} \tau_i^{(j)} f(\theta_i^{(j)}) \right), \quad \tau_i = [e_1^T y_i]^2$$

where $T_m^{(j)}$ represents the result of a degree-$(m+1)$ Krylov expansion of the pair $(A, v^{(j)})$ for any isotropically distributed $v^{(j)} \sim \mathcal{D}$ whose outer product $\otimes$ satisfies $\mathbb{E}[v^{(j)} \otimes v^{(j)}] = I$, and (θ_i, y_i) represent the Rayleigh-Ritz pairs associated with the tridiagonal eigendecomposition $T_m = Y \Theta Y^T$. Note that setting $f(\lambda) = \mathrm{sgn}_+(\lambda)$ yields $\mathrm{tr}(f(A)) = \sum_{i=1}^{n} \mathrm{sgn}_+(\lambda_i) = \mathrm{rank(A)}$, motivating the use of trace estimation methods for both rank and spectral sum approximation. 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 $O(\epsilon^{-2} \log(\eta^{-1}))$ evaluations of $e_1^T f(T_m) e_1$, where each T_m is generated by applying degree-$(m+1)$ Lanczos to pairs (A, v) generated over random $v \sim \mathcal{D}$. This probabilistic 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 at a small additional re-orthogonalization and storage costs.

4.3 Apparent pairs optimization

One of the defining aspects of the computation is that, unlike the reduction algorithm, execution does not require matrix decomposition. While this distinction is important, it does not necessarily bar the transfer of persistence-specific optimizations to our matvec-based approach. One such optimization applicable to clique filtrations—the identification of *apparent pairs*—enables us to discard upwards of 90-99% of the columns of ∂_p prior to any rank computations.

Apparent pairs (APs) are a class of persistence pairs that are already reduced in the filtration boundary matrix $\partial(K)$. For simplexwise clique filtrations, the main advantage of APs is that they can be readily identified based on a purely local condition, which is evident from their very definition:

Definition 2. Given a simplexwise filtration $K_\preceq$ of a simplicial complex K, a pair of simplices (τ, σ) of K is called an *apparent pair* of $K_\preceq$ if (1) τ is youngest facet of σ, and (2) σ is the oldest cofacet of τ.

[8] Recall that if $A \in \mathbb{R}^{n \times n}$ has eigenvalue decomposition $A = U \Lambda U^T$, the matrix function $f(A) \in \mathbb{R}^{n \times n}$ is $U f(\Lambda) U^T$, where $f(\Lambda)$ applies f to each diagonal entry of Λ.

Equivalently, a pair (τ, σ) of K is *apparent* if all entries below or to the left of (τ, σ) in the filtered boundary matrix ∂ are zero. Note that any persistent pair $(\tau, \sigma) \in \mathrm{dgm}_p(K)$ corresponds to a pair of columns in R, one of which is entirely zero and one recording a non-zero pivot entry satisfying (2.4). From a rank-based perspective, this suggests we may safely discard $p+1$ simplices whose boundary chains lie in the kernel of R. We formalize this with a lemma.

Lemma 4. *Let $\partial_{p+1}(K)$ denote the dimension $p+1$ filtered boundary matrix obtained from the $R = \partial V$ decomposition of a simplexwise filtration (K, f). Then, the p-th persistence diagram $\mathrm{dgm}_p(K)$ determines a partitioning of $\partial_{p+1}(K) = [\, \partial^-_{p+1} \mid \partial^+_{p+1} \,]$ satisfying:*

$$\mathrm{rank}\big(\partial_{p+1}(K)\big) = \mathrm{rank}\big(\partial^-_{p+1}(K)\big)$$

where the partitioning of ∂_{p+1} is determined by the location of the pivot entries in R:

$$\partial^-_{p+1} = \{\, \partial[\sigma_j] : \mathrm{col}_R(j) \neq 0, \sigma_j \in K^{p+1} \,\}, \quad \partial^+_{p+1} = \{\, \partial[\sigma_j] : \mathrm{col}_R(j) = 0, \sigma_j \in K^{p+1} \,\}$$

The practical significance of Lemma 4 is that, prior to any rank computations, we may detect and remove simplices whose elementary boundary chains do not affect the overall rank of the operator. In particular, though determining ∂^-_{p+1} exactly requires constructing the full decomposition $R = \partial K$, an approximation $\hat{\partial}^-_{p+1} \supset \partial_{p+1}$ can be determined by identifying apparent pairs in $(\sigma, \eta) \in \mathrm{dgm}_{p+1}(K)$—the corresponding boundary chains $\partial[\sigma]$ by definition must lie in ∂^+_{p+1}, and therefore can be removed. Note that the set of APs does not depend on the coefficient field the homology of the complex is defined over, though they do depend on the simplexwise ordering.

Identifying an AP $(\tau, \sigma) \in \mathrm{dgm}_p(K)$ can be reduced to enumerating cofacets of $\tau \in K^p$. To do this efficiently, we follow the low-memory approach used in the popular software Ripser [2], which restricts its computation to simplexwise filtrations induced by the *reverse colexicographical* vertex order. In this setting, p-simplices $(\, v_{i_{p+1}}, \ldots, v_{i_0} \,)$ satisfying $v_{i_{p+1}} > \cdots > v_{i_0}$ are mapped to integers $r \in [0, \binom{n}{p+1})$ via a correspondence in the *combinatorial number system*; when the colexicographical order is used, the bijection is given by the sum $(\, v_{i_{p+1}}, \ldots, v_{i_0} \,) \mapsto \sum_{j=1}^{p+1} \binom{i_j}{j}$, and cofacets are given by the following relation:

$$(v_{i_{p+1}}, \ldots, v_{i_{k+1}}, v_j, v_{i_k}, \ldots, v_{i_0}) \mapsto \sum_{l=k+1}^{p+1} \binom{i_l}{l+2} + \binom{j}{k+1} + \sum_{l=0}^{k} \binom{i_l}{l+1} \tag{4.1}$$

By enumerating $j = n-1, \ldots, 0$ for $j \notin [n] \setminus \{i_p, \ldots, i_0\}$, one recovers all of $n-(p+1)$ cofacets of $(\, v_{i_{p+1}}, \ldots, v_{i_0} \,)$ in reverse colexicographic vertex order. Cofacet enumeration in the colexicographical order is particularly efficient due to the fact that the left and right partial sums in (4.1) can be maintained throughout the enumeration: assuming all $O(n \cdot (p+1))$ binomial coefficients are precomputed, finding all the cofacets of any p-simplex τ requires just $O(n-p)$ additions in integer arithmetic.

The most prevalent subset of APs relevant to the clique-filtrations are those with zero persistence. In practice, zero persistence APs may be identified without enumerating all cofacets by using a "shortcut" involving a correspondence between zero-persistence APs and lexicographically minimal (maximal, respectively) facet (cofacet, respectively) pairs (see Proposition 3.12 in [2]). This "shortcut" is particularly useful due to the fact that zero persistence pairs comprise a large proportion of the apparent pairs—indeed, Theorem 3.10 in [2] shows that in dimension 1, the zero persistence pairs of a simplexwise refinement of the Vietoris-Rips filtration are *precisely* the apparent pairs of the same filtration.

To demonstrate this, we computed the number of apparent pairs in a variety of common data sets, including their apparent positive and negative

Dataset	n	p	$\lvert K^p \rvert$	# apparent (+/-)	# non-apparent
SW1Pers	50	1	1,068	1,004 (49/954)	64
	50	2	13,300	13,051 (12,048/1,003)	249

Note that positive apparent cofacets in dimension $p+1$ may be removed entirely from the computation of μ^*_p and β^*_p, prior to performing any matvecs. On the contrary, we cannot remove negative apparent cofacets from the dimension $p+1$ boundary operators, as they may (or may not) have linear dependents—their effect

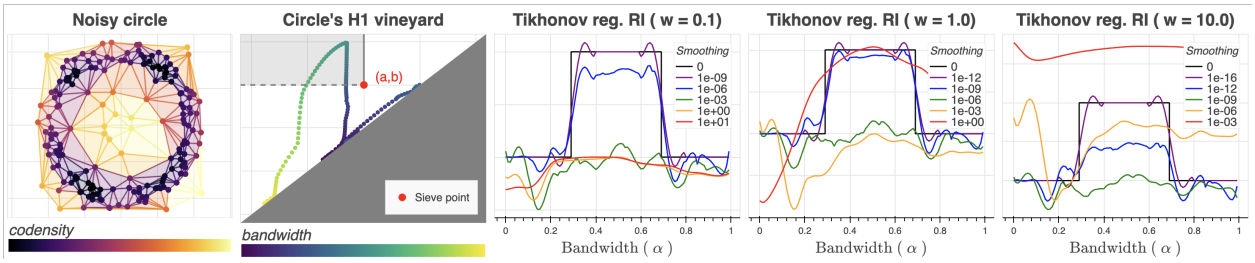


Figure 3: From left to right: Delaunay complex K realized from point set $X \subset \mathbb{R}^2$ sampled with multiple types of noise around S^1 (colored by codensity at optimal $\alpha^* \approx 1/2$); codensity vineyard of (K, f_α) across varying bandwidths α and a fixed sieve point $(a, b) \in \Delta_+$; 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.

on the rank of the constitutive boundary terms can only determined through reduction. On the positive side, the vast majority of apparent pairs in dimension $p+1$ must be positive, for the number of negative such pairs is bounded by $O(|K^p|)$. Moreover,

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 to employing persistence, one often removes outliers from point set $X \subset \mathbb{R}^d$ via some density-based pruning heuristic that itself is parameterized. This is typically necessary due to the fact that, though stable under Hausdorff noise [14], diagrams are notably unstable against *strong outliers*—even one point can ruin the summary. As an exemplary use-case of our spectral-based method, we re-cast the problem of identifying strong outliers below 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 appropriately chosen $(a, b) \in \Delta_+$ over the pair (K, f_α), where $f_\alpha : X \to \mathbb{R}_+$ is a kernel (co)-density estimate:

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

where $C(\mathcal{K}_\alpha)$ is a normalizing constant that depends on the bandwidth-parameterized kernel, $\mathcal{K}_\alpha$. Intuitively, if there exists a choice of bandwidth α^* which distinguishes strong outliers from Hausdorff noise clustered around S^1, then that choice of bandwidth should exhibit a highly persistent pair $(a^*, b^*) \in \mathrm{dgm}_1(K, f_{\alpha^*})$. If the corner point (a, b) captures this occurrence and $|a - b|$ is large enough, we expect $\beta_p^{a,b}(\alpha) = 1$ near the optimal bandwidth α^*—matching the first Betti number of S^1—and 0 otherwise.

In Figure 3, we depict the vineyard of dgm_1 of a simple Delaunay complex and codensity pair (K, f_α), along with the sieve point (a, b) and the region wherein S^1 is accurately captured by persistence. 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. By using the Tikhonov regularization from (3.15), we obtain continuously varying objective curves from $\hat{\beta}_p^{a,b}(\alpha; \tau)$ which are guaranteed to have the same maxima as $\beta_p^{a,b}(\alpha)$ as $\tau \to 0$, as shown in Figure 3. Practical optimization of these types of objective surfaces can be handled via *iterative thresholding*, a technique which alternates between gradient steps to reduce the objective and thresholding steps to enforce the rank constraints []. We leave the tuning of such optimizers to future work.

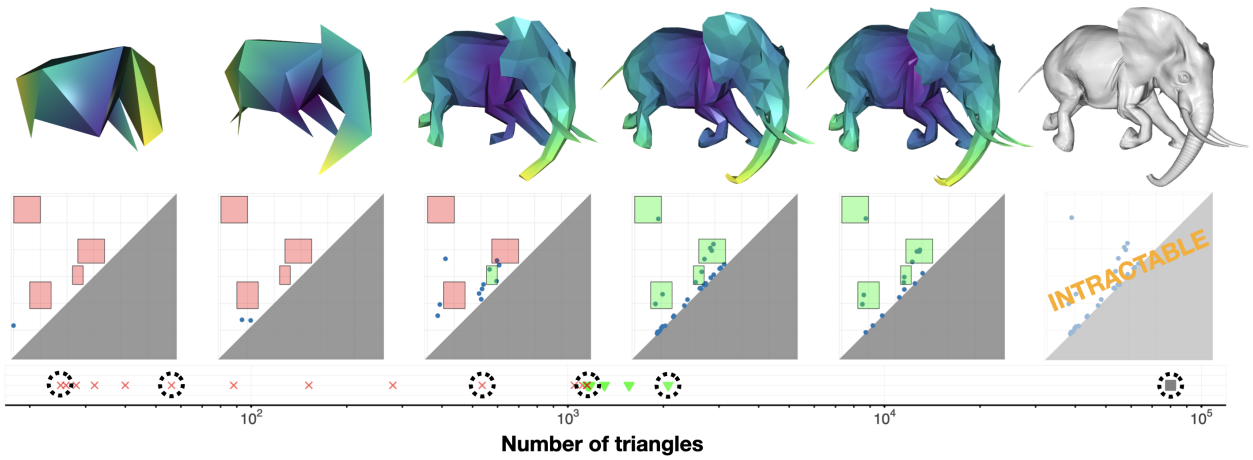


Figure 4: (Top) Meshes filtered and colored by eccentricity at varying levels of simplification; (middle) their diagrams and topological constraints; (bottom) simplification thresholds tested by an exponential search, on a logarithmic scale. The color/shape of the markers indicate whether the corresponding 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.

Topology-guided simplification

In many 3D computer graphics applications, one would like to simplify a given simplicial or polygonal mesh embedded in $\mathbb{R}^3$ so as to decrease its level of detail (LOD) while retaining its principal geometric structure(s). Such simplifications are often necessary to improve the efficiency of compute-intensive tasks that depend on the size of the mesh (e.g. rendering). Though many simplification methods developed to preserve geometric criteria—such as curvature, co-planarity, or distance—are now well known (see [23] for an overview), by comparison *topology-preserving* simplification procedures are relatively sparse, especially for higher embedding dimensions. Moreover, many topology-preserving procedures are developed by restricting to operations that preserve *local* notions of topology, such as the genus of a feature's immediate neighborhood; such procedures are known to greatly limit the amount of detail decimation algorithms can remove.

As a prototypical application of our proposed relaxation, we re-visit the mesh simplification problem under *persistence-based* constraints, as is done in e.g. [20]. In contrast to [20], however, we forgo the use of local persistence-preserving operations and instead opt for a simpler strategy: we assume the set of topological constraints are given as sieve, and we

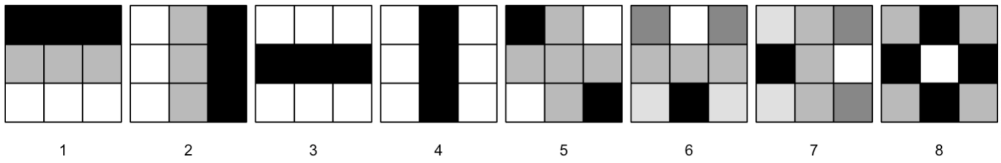
Manifold detection from image patches

A common hypothesis is that high dimensional data tend to lie in the vicinity of an embedded, low dimensional manifold or topological space. An exemplary demonstration of this is given in the analysis by Lee et al. [27], who explored the space of high-contrast patches extracted from Hans van Hateren's still image collection[9], which consists of $\approx 4,000$ monochrome images depicting various areas outside Groningen (Holland). Originally motivated by discerning whether there existed clear qualitative differences in the distributions of patches extracted from images of different modalities, such as optical and range images, Lee et al. [27] were interested in exploring how high-contrast 3×3 image patches were distributed in pixel-space with respect to predicted spaces and manifolds. Formally, they measured contrast using a discrete version of the scale-invariant Dirichlet semi-norm:

$$\|x\|_D = \sqrt{\sum_{i \sim j} (x_i - x_j)^2} = \sqrt{x^T D x}$$

[9]See `http://bethgelab.org/datasets/vanhateren/` for details on the image collection.

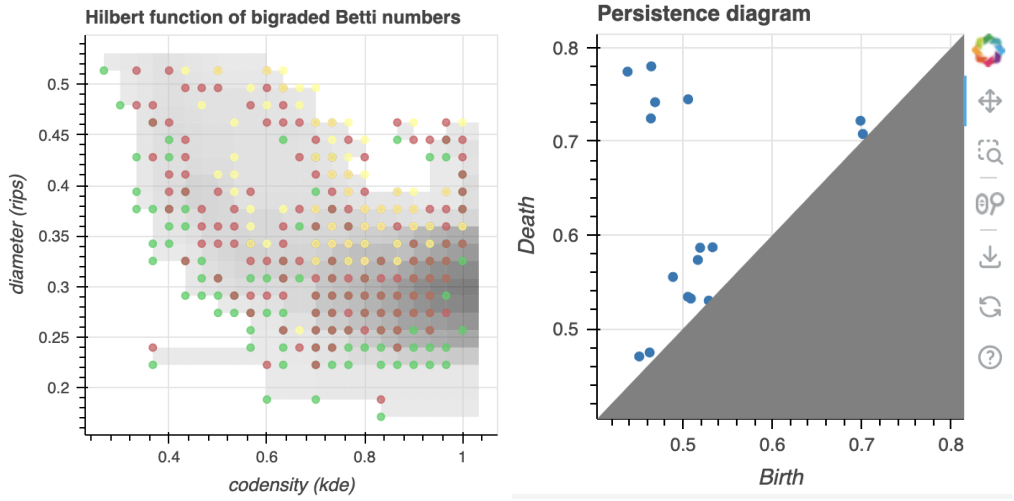
where D is a fixed matrix whose quadratic form $x^T Dx$ applied to an image $x \in \mathbb{R}^9$ is proportional to the sum of the differences between each pixels 4 connected neighbors (given above by the relation $i \sim j$). Their research was . By mean-centering, contrast normalizing, and"whitening" the data via the Discrete Cosine Transform (DCT), they show a convenient basis for D may be obtained via an expansion of 8 certain non-constant eigenvectors:



Since these images are scale-invariant, the expansion of these basis vectors span the 7-sphere, $S^7 \subset \mathbb{R}^8$. Using a Voronoi cell decomposition of the data, their distribution analysis suggested that the majority of data points concentrated in a few high-density regions.

In follow-up work, Carlsson et al. [8] used persistent homology to find the distribution of high-contrast 3×3 patches is actually well-approximated by a Klein bottle $\mathcal{M}$—around 60% of the high-contrast patches from the still image data set lie within a small neighborhood around $\mathcal{M}$ accounting for only 21% of the 7-sphere's volume. Though a certainly remarkable result, if one was not aware of the analysis done by [27, 8], it would not be immediately clear a priori how to reproduce the discovery in the more general setting; e.g. how does one determine whether the Klein bottle model is a good candidate space for characterizing the space of these image patches? Indeed, armed with a refined topological intuition, Carlsson still needed to perform extensive sampling, preprocessing, and model fitting techniques in order to reveal the underlying topological space with persistent homology [8]. Unfortunately, such preprocessing is necessary is due to persistent homology's aforementioned instability with respect to strong outliers.

In contrast to the 1-parameter setting, a multi-parameter persistence approach that accounts for the local density of points to be able to identify the presence of the Klein bottle, even in the midst of outliers.



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 [12]) 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. Ripser: efficient computation of vietoris–rips persistence barcodes. *Journal of Applied and Computational Topology*, 5(3):391–423, 2021.

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

[4] 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.

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

[6] 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.

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

[8] Gunnar Carlsson, Tigran Ishkhanov, Vin De Silva, and Afra Zomorodian. On the local behavior of spaces of natural images. *International journal of computer vision*, 76:1–12, 2008.

[9] 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.

[10] 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.

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

[12] 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.

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

[14] 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.

[15] 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.

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

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

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

[19] 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.

[20] Ulderico Fugacci, Michael Kerber, and Hugo Manet. Topology-preserving terrain simplification. In *Proceedings of the 28th International Conference on Advances in Geographic Information Systems*, pages 36–47, 2020.

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

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

[23] Paul S Heckbert and Michael Garland. Survey of polygonal surface simplification algorithms. Siggraph, 1997.

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

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

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

[27] Ann B Lee, Kim S Pedersen, and David Mumford. The nonlinear statistics of high-contrast patches in natural images. *International Journal of Computer Vision*, 54:83–103, 2003.

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

[29] 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.

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

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

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

[33] 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.

[34] 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.

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

[36] 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.

[37] 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.

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

[39] Shang-Hua Teng. The laplacian paradigm: Emerging algorithms for massive graphs. In *Theory and Applications of Models of Computation: 7th Annual Conference, TAMC 2010, Prague, Czech Republic, June 7-11, 2010. Proceedings 7*, pages 2–14. Springer, 2010.

[40] 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.

[41] Shashanka Ubaru and Yousef Saad. Fast methods for estimating the numerical rank of large matrices. In *International Conference on Machine Learning*, pages 468–477. PMLR, 2016.

[42] 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.

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

A Appendix

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:

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

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 []:

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

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:

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

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 $W_p = \text{diag}(\{ w(\sigma_i) \}_{i=1}^n)$ 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 [29], L_p^{up} may be written as follows:

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

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:

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

where $\mathcal{A}_p^{\text{up}}$ is a weighted adjacency matrix, and $\mathcal{I}_n$ is the identity matrix with $\mathcal{I}(\tau) = \text{sign}(w(\tau))$ (see Section **??**). The primary benefit of this normalization is that it guarantees $\Lambda(\mathcal{L}_p^{\text{up}}) \subseteq [0, p+2]$ 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 $\mathcal{L}_p^{\text{up}}$ 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 $D = \text{diag}(\{\deg(v_i)\})$ 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 [13], 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 $\text{co}(\tau) = \{\, \sigma \in K^{p+1} \mid \tau \subset \sigma \,\}$ the set of proper cofaces of $\tau \in K^p$, or *cofacets*, and the (weighted) *degree* of $\tau \in K^p$ with:

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

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 $\{\sigma\} = \text{co}(\tau) \cap \text{co}(\tau')$ is in fact *unique* [21]. Thus, we may use a relation $\tau \overset{\sigma}{\sim} \tau'$ to rewrite the operator from (A.7) element-wise:

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

where $s_{\tau,\tau'} = \text{sgn}([\tau], \partial[\sigma]) \cdot \text{sgn}([\tau], \partial[\sigma])$. 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:

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

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(s) $N = |K|$, $n = |K^p|$, $m = |K^{p+1}|$
Optional: Weight arrays $w_{p+1} \in \mathbb{R}^m_+$ and $w_p \in \mathbb{R}^n_+$
Output: $y = \langle L^{\text{up}}_p, x \rangle = (W_p \circ \partial_{p+1} \circ W_{p+1} \circ \partial^T_{p+1} \circ W_p)x$

```
1:  // Precompute weighted degrees deg_w
2:  Define h : K^p → [n]
3:  deg_w ← 0
4:  for σ ∈ K^{p+1}, k ∈ [m] do:
5:      for τ ∈ ∂[σ] do:
6:          deg_w[h(τ)] ← deg_w[h(τ)] + w_p[h(τ)] · w_{p+1}[k] · w_p[h(τ)]
7:
8:  function UpLaplacianMatvec(x ∈ R^n)
9:      y ← deg_w ⊙ x   (element-wise product)
10:     for σ ∈ K^{p+1}, k ∈ [m] do:
11:         for τ, τ' ∈ ∂[σ] × ∂[σ] where τ ≠ τ' do:
12:             s_{τ,τ'} ← sgn([τ], ∂[σ]) · sgn([τ'], ∂[σ])
13:             i, j ← h(τ), h(τ')
14:             y_i ← y_i + s_{τ,τ'} · x_j · w_p[i] · w_{p+1}[k] · w_p[j]
15:     return y
```

In general, the signs of the coefficients $\text{sgn}([\tau], \partial[\sigma])$ and $\text{sgn}([\tau'], \partial[\sigma])$ depend on the position of τ, τ' as summands in $\partial[\sigma]$, 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 [21])—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.

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 [19]. 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 [19, 3, 16] for details.

Given a filtration (K, f) of size $m = |K|$ with filter $f : K \to [m]$, the reduction algorithm in form given in [19] 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 [16]), 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 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 [16]. 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 $\approx 98 - 99\%$ [3]. 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 [3] 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 [16], 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 [30])**.** 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 [30].

A.4 Examples of Parameterized 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...

[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 $t \in \mathbb{R}$.

A.5 Proofs

Proof of Lemma 1

Proof. The Pairing Uniqueness Lemma [16] 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:

$$\text{low}_R[j] = i \iff r_R(i,j) \neq 0 \iff r_\partial(i,j) \neq 0 \tag{A.14}$$

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:

$$\dim(Z_p(K_i) \cap B_p(K_i)) = |\Gamma_p^{i,j}| \tag{A.15}$$

where $\Gamma_p^{i,j} = \{\, \text{col}_{R_{p+1}[k]} \neq 0 \mid \; k \in [j],\, 1 \leq \text{low}_{R_{p+1}}[k] \leq i \,\}$. 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 $\text{low}_{R_{p+1}}[k] > i$. Combining rank-nullity with the observation above, we have:

$$|\bar{\Gamma}_p^{i,j}| = \dim(B_p(K_j)) - |\Gamma_p^{i,j}| = \text{rank}(R_{p+1}^{i+1,j}) \tag{A.16}$$

Combining equations (A.15) and (A.16) yields:

$$\dim(Z_p(K_i) \cap B_p(K_j)) = |\Gamma_p^{i,j}| = \dim(B_p(K_j)) - |\bar{\Gamma}_p^{i,j}| = \text{rank}(R_{p+1}^{1,j}) - \text{rank}(R_{p+1}^{i+1,j}) \tag{A.17}$$

Observing the final matrices in (A.17) 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 $K_t = \text{Rips}_\epsilon(\delta_{\mathcal{X}}(t))$ with non-zero entries satisfying (**??**) by definition yields a matrix ∂_p satisfying $\text{rank}(\partial_p) = \dim(\text{B}_{p-1}(K_t))$. 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 $\text{diam}(\sigma) < \epsilon$. Thus, $\text{rank}(\partial_p^t) = \dim(\text{B}_{p-1}(K_t))$ 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.6 Proofs of basic properties

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