1 Introduction

Motivation

TODO

Contributions: In this effort, we introduce an iterative ϵ-approximation method for computing the *the persistent counting invariants*—μ_p^* and β_p^*—in *essentially* $O(m)$ memory and $O(mn)$ time, where m, n are the number of $p+1, p$ simplices in the complex, respectively. The approximation method is spectral-based and is particularly efficient when frequent recomputation is needed on perturbed inputs, which we generically refer to as the *parameterized setting.* We show that, when the accuracy parameter ϵ is made small enough, both invariants are recovered exactly. In deriving the approximation, we naturally obtain continuous variations of the counting invariants which are smooth and differentiable on the positive semi-definite cone, which we believe to be of independent interest. One particular advantage of the method is that, unlike other dynamic or "kinematic" computations, it does not require the complex K be explicitly filtered nor does it require any complicated data structure maintenance procedures. Indeed, as algorithm does not modify the boundary matrix $\partial(K)$ at all, it need not necessarily have $\partial(K)$ stored in memory at all. Interestingly, our results also imply the existence of an subcubic, output-sensitive algorithm for computing persistence diagrams (via [6]) that inherits all of the aforementioned benefits and requires as its only input an boundary matrix-vector operator $x \mapsto \partial x$.

1.1 Organization

Section 2 introduces the notation and background theory on which the rest of the paper depends. Included are some theoretical motivations for this work in subsection 2, an illuminating derivation of the PBN in 2, a time-varying parameterization of boundary matrix, and a brief recount of related work in persistence-related algorithms. Section **??** contains the main results: the spectral relaxation of the counting invariants, the complexities of computing them, and their basic properties.

Finally, we illustrate both the computational and practical advantages our expression has with a few applications in section 6. Some technical details, illustrative examples, and pseudocode are relegated to the appendix for readability.

2 Background & Notation

A *simplicial complex* $K \subseteq \mathcal{P}(V)$ over a vertex set $V = \{v_1, v_2, \dots, v_n\}$ is a collection of simplices $\{\sigma : \sigma \in \mathcal{P}(V)\}$ such that $\tau \subseteq \sigma \in K \implies \tau \in K$. We denote with $K^p = \{\sigma \in K : \dim(\sigma) = p\}$ the p-simplices of K and by $K^{(p)} = \{\sigma \in K : \dim(\sigma) \leq p\}$ the p-skeleton of K. A *filtration* $K_\bullet = \{K_i\}_{i \in I}$ of a simplicial complexes indexed by a totally ordered set I is a family of complexes such that $i < j \in I \implies K_i \subseteq K_j$. $K_\bullet$ is called *simplexwise* if $K_j \setminus K_i = \{\sigma_j\}$ whenever j is the immediate successor of i in I and $K_\bullet$ is called *essential* if $i \neq j$ implies $K_i \neq K_j$:

$$\emptyset = K_0 \subsetneq K_1 \subsetneq \cdots \subsetneq K_m = K_\bullet, \quad K_i = K_{i-1} \cup \{\sigma_i\} \tag{2.1}$$

Filtrations may be equivalently defined via *filter functions* $f : K \to I$ satisfying $f(\tau) \leq f(\sigma)$ whenever $\tau \subseteq \sigma$. Here, we consider two index sets for I: $\mathbb{R}$ and $[n] = \{1, \dots, n\}$. Any finite filtration may be trivially converted into an essential, simplexwise filtration via a set of *condensing*, *refining*, and *reindexing* maps [1]. For simplicity, but without loss of generality, we exclusively consider essential simplexwise filtrations and for brevity refer to them as filtrations.

Given a simplicial complex $K \subseteq \mathcal{P}(V)$ and a strictly increasing subset $\sigma = \{v_1, v_2, \dots, v_{p+1}\} \subseteq V$ satisfying $v_1 < v_2 < \cdots < v_{p+1}$, an *oriented p-simplex* $[\sigma] = [v_1, v_2, \dots, v_{p+1}]$ is defined as:

$$[\sigma] = (-1)^{|\pi|} \left[v_{\pi(1)}, v_{\pi(2)}, \dots, v_{\pi(p+1)} \right] \tag{2.2}$$

where π is a permutation on $[\, p+1\,]$ and $|\pi|$ is the number of inversions of that permutation. The The p-boundary ∂_p of an oriented p-simplex $[\sigma] \in K$ is defined as the alternating sum of its oriented co-dimension 1 faces:

$$\partial_p[\sigma] := \sum_{i=1}^{p+1} (-1)^{i-1} [v_1, \dots, v_{i-1}, v_{i+1}, \dots v_{p+1}] \tag{2.3}$$

The concepts of boundaries and orientation generalizes beyond simplices. Given a field $\mathbb{F}$, an *oriented p-chain* is a formal $\mathbb{F}$-linear combination of oriented p-simplices of K, the collection on which under addition yields an $\mathbb{F}$-vector space denoted $C_p(K)$. Given a p-chain $c \in C_p(K)$, its p-boundary $\partial_p[c]$ is defined linearly in terms of its constitutive simplices; chains satisfying $\partial_p[c] = 0$ are called *p-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. Since $\partial_p \circ \partial_{p+1} = 0$ for all $p \geq 0$, the quotient space $H_p(K) = Z_p(K)/B_p(K)$ is a well-defined group called the *p-th homology group of K* with coefficients in $\mathbb{F}$. The dimension of the p-th homology group $\beta_p(K) = \dim(H_p(K))$ of K is called the *p-th Betti number* of K.

Let $K_\bullet = \{K_i\}_{i \in [m]}$ denote a filtration of size $|K_\bullet| = m$, and let $\Delta_+^m = \{(i,j) : 0 \leq i \leq j \leq m\}$ denote the set of filtration index pairs. For every such pair $(i,j) \in \Delta_+^m$, the inclusions $K_i \subsetneq K_{i+1} \subsetneq \cdots \subsetneq K_j$ induce linear transformations $h_p^{i,j}$ at the level of homology:

$$0 = H_p(K_0) \to \cdots \to H_p(K_i) \underbrace{\to \cdots \to}_{h_p^{i,j}} H_p(K_j) \to \cdots \to H_p(K_m) = H_p(K_\bullet) \tag{2.4}$$

When $\mathbb{F}$ is a field, this sequence of homology groups uniquely decomposes $K_\bullet$ into a pairing of simplices (σ_i, σ_j) demarcating the evolution of homology classes [20]: σ_i marks the creation of a homology class, σ_j marks its destruction, and the difference $|i - j|$ records the lifetime of the class, called its *persistence*. The p-th persistent homology groups are the images of these transformations and the p-th persistent Betti numbers are their dimensions:

$$H_p^{i,j} = \begin{cases} H(K_i) & i = j \\ \operatorname{Im} h_p^{i,j} & i < j \end{cases}, \qquad \beta_p^{i,j} = \begin{cases} \beta_p(K_i) & i = j \\ \dim(H_p^{i,j}) & i < j \end{cases} \tag{2.5}$$

For a fixed $p \geq 0$, the collection of persistent pairs (i,j) together with unpaired simplices (l, ∞) form a summary representation $\mathrm{dgm}_p(K_\bullet)$ called the *p-th persistence diagram of $K_\bullet$*. Conceptually, $\beta_p^{i,j}$ counts the number of persistent pairs lying inside the box $(-\infty, i] \times (j, \infty)$ (see Figure 1)—the number of persistent homology groups born at or before i that died sometime after j.

The duality between PBNs and Diagrams

The connection between the persistent homology (PH) groups and their corresponding persistent Betti numbers (PBNs) has long been studied from multiple perspectives by several authors [4, 5, 9, 20]. From an algebraic perspective, Carlsson et al. [20] observed that the PH groups over a filtration may be viewed as the standard homology groups of a particular graded module M over a polynomial ring. In [9], Cohen-Steiner et al. give a more discrete perspective on PH by defining the persistence diagram in terms of a *multiplicities*: given a tame function $f : \mathcal{X} \to \mathbb{R}$ over a topological space $\mathcal{X}$, its homological critical values $\{a_i\}_{i=1}^n$, and an interleaved sequence $\{b_i\}_{i=0}^n$ satisfying $b_{i-1} < a_i < b_i$ for all $1 \leq i \leq n$, the p-th persistence diagram over f is given as:

$$\mathrm{dgm}_p(f) = \{\, (a_i, a_j) : \mu_p^{i,j} \neq 0 \,\} \,\cup\, \Delta \tag{2.6}$$

where Δ denotes the diagonal, counted with infinite multiplicity, and $\mu_p^{i,j}$ is the *multiplicity function*, defined as:

$$\mu_p^{i,j} = \left(\beta_p^{i,j\text{-}1} - \beta_p^{i,j}\right) - \left(\beta_p^{i\text{-}1,j\text{-}1} - \beta_p^{i\text{-}1,j}\right) \qquad \text{for } 0 \leq i < j \leq n+1 \tag{2.7}$$

Equation (2.7) illuminates an intrinsic connection between the multiplicity function and the persistent Betti numbers. Namely, the inclusion-exclusion property that (2.7) obeys suggests that diagrams completely characterize their PBNs. Indeed, the fundamental lemma of persistent homology [12] states that for every pair of indices $0 \leq k \leq l \leq n+1$:

$$\beta_p^{k,l} = \sum_{i \leq k} \sum_{j > l} \mu_p^{i,j} \tag{2.8}$$

The direct consequence of (2.8) is that if one is interested in computing any of the PBNs of some space $\mathcal{X}$, then it is sufficient to compute $\mathrm{dgm}_p(\mathcal{X})$ and read them off directly. In this effort, as we will show, we will take the inverse mentality by recovering portions of the diagram via multiplicity evaluations.

The duality between diagrams and PBNs derived from persistence modules indexed over the real line was further studied by Chazal [5] in a measure-theoretic setting. By reinterpreting the multiplicity function μ_p^* as a certain kind of integer-valued measure over rectangles in the plane, a generalization of (2.7) was shown by demonstrating that one may recover the diagram of a persistence module M over $\mathbb{R}$ by constructing its corresponding *persistence measure*:

$$\mu_p(R; M) = \operatorname{card}\left(\mathrm{dgm}_p(M)\big|_R \right) \qquad \text{for all rectangles } R \subset \mathbb{R}^2 \tag{2.9}$$

Cerri et al. [4] incorporate this interpretation in their work studying the stability of PBNs in multidimensional persistence by showing that *proper cornerpoints* in the persistence diagram are points $x = (i, j) \in \Delta_+$ satisfying:

$$x = (i, j) \in \mathrm{dgm}_p(f) \iff \mu_p(x) > 0 \iff \min_{\delta>0} \left(\beta_p^{i+\delta,j-\delta} - \beta_p^{i+\delta,j+\delta}\right) - \left(\beta_p^{i-\delta,j-\delta} - \beta_p^{i-\delta,j+\delta}\right) > 0 \tag{2.10}$$

One may compare (2.7) with (2.10). One of the primary contributions from [4] is a representation theorem akin to (2.8) (Theorem 3.11) expressing the persistent Betti number function $\beta_* : \Delta_+ \to \mathbb{N} \cup \{\infty\}$ as a sum of multiplicity functions. A consequence of this theorem is that distances between diagrams induces a distance between PBN functions—if X is a triangulable space and $f, g : X \to \mathbb{R}$ are two continuous functions, then $d(\beta_f, \beta_g) \leq \max_{x\in X} |f(x) - g(x)|$, where:

$$d(\beta_f, \beta_g) = \inf_{\phi} \sup_{p\in\mathrm{dgm}(f)} \|p - \phi(p)\|_{\tilde\infty}$$

is the (extended) matching distance between PBN functions (β_f, β_g), ϕ ranges over all multi-bijections between $\mathrm{dgm}(f)$ and $\mathrm{dgm}(g)$, and $\|\cdot\|_{\tilde\infty}$ measures the pseudo-distance [4] between points. Thus, PBN functions are stable functions: small changes in continuous scalar-valued filtering functions imply small changes in the corresponding persistent Betti numbers functions.

Like persistence diagrams, the stability of PBN functions justifies their use and study in continuously parameterized settings. Despite this, much of persistence-related research has concentrated on exploiting properties of the diagram itself [], as opposed to the PBNs ([4, 6] are notable exceptions). Nonetheless, as we shall show, there are certain advantages the PBN computation has over the "standard" reduction algorithm from [12]. Indeed, (2.7) implies one may in theory recover the diagram through [a finite number of] PBN computations alone via a divide-and-conquer like approach [6], suggesting an alternative computational paradigm—distinct from the reduction family of algorithms—with which to approach the persistence computation.

2.1 Motivating Derivation

Let $B_p(K_\bullet) \subseteq Z_p(K_\bullet) \subseteq C_p(K_\bullet)$ denote the p-th boundary, cycle, and chain groups of a given filtration $K_\bullet$, respectively. Additionally, let $\partial_p : C_p(K_\bullet) \to C_{p-1}(K_\bullet)$ denote the boundary operator sending p-chains to their respective boundaries. With a slight abuse of notation, we also use ∂_p to also denote the p-th filtration boundary matrix with respect to an ordered basis $(\sigma_i)_{1\leq i\leq N}$ induced by $K_\bullet$:

$$\partial_p[i, j] = \begin{cases} (-1)^{s_{ij}} & \sigma_i \in \partial_p[\sigma_j] \\ 0 & \text{otherwise} \end{cases} \tag{2.11}$$

As the matrix representation of the p-th boundary operator $\partial_p : C_p(K_\bullet) \to C_p(K_\bullet)$, note that ∂_p is completely characterized by the oriented p and $p-1$ simplices of $K_\bullet$. The p-th persistent Betti number $\beta_p^{i,j}$ at index $(i, j) \in \Delta_+^N$, where $\Delta_+^N := \{\, (i, j) \in [N] \times [N] : i < j \,\}$, is defined as:

$$\begin{aligned} \beta_p^{i,j} = \dim(H_p^{i,j}) &= \dim\left(Z_p(K_i)/B_p(K_j)\right) \\ &= \dim\left(Z_p(K_i)/(Z_p(K_i) \cap B_p(K_j)\right) \\ &= \dim\left(Z_p(K_i)\right) - \dim\left(Z_p(K_i) \cap B_p(K_j)\right) \end{aligned} \tag{2.12}$$

While $\dim(Z_p(K_i)) = \mathrm{nullity}(\partial_p(K_i))$ and thus reduces to a rank computation, the intersection term—the persistence part—is more subtle. Zomorodian et al. [20] outline an algorithm to compute a basis for $Z_p(K_i) \cap B_p(K_j)$ via a sequence of boundary matrix reductions; their subsequent Theorem 5.1 reduces the complexity of computing PH groups with coefficients in any PID to that of computing homology groups. However, the standard homology computations require $O(N^2)$ space and $O(N^3)$ time to compute, implying either of these approaches to computing the PBN computation exhibits same complexity as the full persistence computation.

In what follows, we outline a different approach to computing (2.12) that we argue is not only simpler and computationally attractive, but also amenable to acceleration in dynamic settings. To illustrate our approach, we require more notation. If A is a $m \times n$ matrix, let $A^{i,j}$ denote the lower-left submatrix defined by the first j columns and the last $m - i + 1$ rows (rows i through m, inclusive). For any $1 \leq i < j \leq N$, define the quantity $r_A(i, j)$ as follows:

$$r_A(i, j) = \mathrm{rank}(A^{i,j}) - \mathrm{rank}(A^{i+1,j}) + \mathrm{rank}(A^{i+1,j-1}) - \mathrm{rank}(A^{i,j-1}) \tag{2.13}$$

The structure theorem from [20] shows that 1-parameter persistence modules can be decomposed in an *essentially unique* way into indecomposables. Computationally, a consequence of this phenomenon is the Pairing Uniqueness Lemma [10], which asserts that if $R = \partial V$ is the decomposition of the boundary matrix, then:

$$r_R(i, j) \neq 0 \Leftrightarrow R[i, j] \neq 0$$

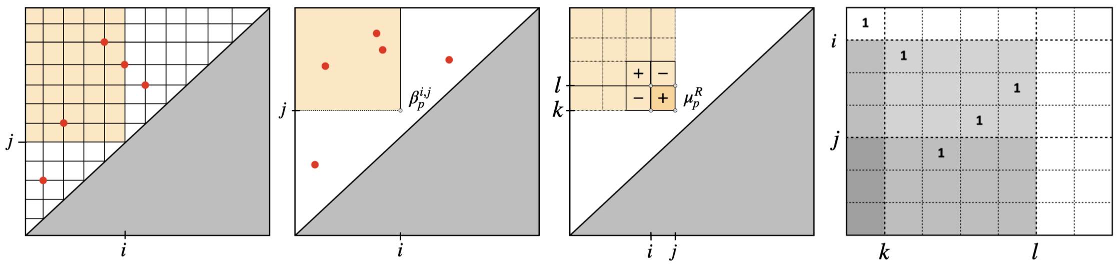

Figure 1: From left to right: $\beta_p^{i,j}$ counts the number of points (3) in upper left-corner of $\mathrm{dgm}_p(K_\bullet)$, where $i, j \in \Delta_+^m$; the same $\beta_p^{i,j}$ with $i, j \in \Delta_+$; the additivity of β_p^* implies μ_p^R over a box $R = [i,j] \times [k,l]$ is given as the sum of four PBNs; generalization of the Pairing Uniqueness Lemma—in this case $\mu_p^R = 4 - 1 - 0 + 0 = 3$ counts pivot entries in the reduced matrix $R = \partial V$.

Since the persistence diagram is derived completely from R, this suggests that information about a diagram can be obtained through rank computations alone. For a more geometric description of this idea, see the third picture in Figure 1. We record a non-trivial fact that follows from this observation:

Lemma 1 (Dey & Wang [11])**.** *Let $R = \partial V$ denote the matrix decomposition of a given filtered boundary matrix ∂ derived from the associated filtration $K_\bullet$. For any pair (i, j) satisfying $1 \leq i < j \leq m$, we have:*

$$\mathrm{rank}(R^{i,j}) = \mathrm{rank}(\partial^{i,j}) \tag{2.14}$$

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

Remark: Though Lemma 1 is defined over the full boundary matrix ∂, there is no loss in generality in its restriction to p-dimensional homology exclusively, for any fixed $p \geq 0$. To see this, note that since the reduction algorithm only adds p-chains to p-chains. Hence, if we set all columns corresponding to simplices of dimension $q \neq p$ to 0 in the $m \times m$ boundary matrix ∂, then ∂ recovers the p-th boundary operator $\partial_p : C_p(K_\bullet) \to C_{p-1}(K_\bullet)$. In what follows, we will use ∂_p and R_p to refer to matrices of ∂ and R whose q-chains are set to 0, for $q \neq p$.

Lemma 1 was the essential motivating step used by Chen et al [6] in their rank-based persistence algorithm—the first output-sensitive algorithm given for computing persistent homology of a filtered complex. The first fact we prove is that Lemma 1 may be used to write the persistent Betti number as a sum of rank functions.

Proposition 1. *Given a fixed $p \geq 0$, a filtration $K_\bullet$ of size $m = |K_\bullet|$, and any pair $(i, j) \in \Delta_+^m$, the persistent Betti number $\beta_p^{i,j}(K_\bullet)$ at (i, j) is given by:*

$$\beta_p^{i,j}(K_\bullet) = i - \mathrm{rank}(\partial_p^{1,i}) - \mathrm{rank}(\partial_{p+1}^{1,j}) + \mathrm{rank}(\partial_{p+1}^{i+1,j}) \tag{2.15}$$

A detailed proof of Proposition 1 is given in the appendix. It turns out Lemma 1 can also be used to generalize (2.15) to arbitrary rectangles in Δ_+ via *μ-queries*: box-parameterized rank queries which count the number of persistence pairs that intersect a fixed "box" placed in the upper half-plane [6]. As a result, the p-th multiplicity function can also be defined in a rank-based formulation akin to (2.15).

Proposition 2 (Chen & Kerber [6])**.** *Given a fixed $p \geq 0$, a filtration $K_\bullet$ of size $m = |K|$, and a $R = [i,j] \times [k,l]$ whose indices (i, j, k, l) satisfy $0 \leq i < j \leq k < l \leq m$, the p-th multiplicity μ_p^R of $K_\bullet$ is given by:*

$$\mu_p^R(K_\bullet) = \mathrm{rank}(\partial_{p+1}^{j+1,k}) - \mathrm{rank}(\partial_{p+1}^{i+1,k}) - \mathrm{rank}(\partial_{p+1}^{j+1,l}) + \mathrm{rank}(\partial_{p+1}^{i+1,l}) \tag{2.16}$$

In summary, the complexity of computing either $\beta_p^{i,j}(K_\bullet)$ or $\mu_p^R(K_\bullet)$ can be reduced to the complexity of computing the rank of a set of "lower-left" submatrices of ∂. We summarize this fact with two corollaries.

Corollary 1. *Given a filtration $K_\bullet$ of size $m = |K_\bullet|$ and indices $i, j \in \Delta_+^m$, computing $\beta_p^{i,j}$ using expression* (2.15) *requires $O(\mathrm{R}_p(j))$ time, where $\mathrm{R}_p(k)$ denotes the complexity of computing the rank of square $k \times k$ matrix with $O((p + 2)k)$ non-zero $\mathbb{F}$ entries.*

Observe the relation $\partial_{p+1}^{i+1,j} \subseteq \partial_{p+1}^{1,j}$ implies the dominant cost of computing (2.15) lies in computing either $\mathrm{rank}(\partial_p^{1,i})$ or $\mathrm{rank}(\partial_{p+1}^{1,j})$. As a result, we obtain a tighter bound for computing μ_p^R that is localized to the pair (K_i, K_l).

Corollary 2. *Given a filtration $K_\bullet$ of size $m = |K_\bullet|$ and a rectangle $R = [i, j] \times [k, l]$ with indices $0 \leq i < j \leq k < l \leq m$, computing μ_p^R using expression* (2.16) *requires $O(\mathrm{R}_p(l - i))$ time, where $\mathrm{R}_p(k)$ denotes the complexity of computing the rank of square $k \times k$ matrix with $O((p + 2)k)$ non-zero $\mathbb{F}$ entries.*

Compared to other classical methods of obtaining $\beta_p^*(K_\bullet)$ and $\mu_p^*(K_\bullet)$, such as those in [12, 20], the primary advantage the rank-based expressions from (2.15)-(2.16) have is that their computations are performed directly on *unfactored* boundary matrices. Moreover, if we only concern ourselves with persistence modules with real-valued coefficients, not only do we inherit the measure-theoretic perspective from 2.9, but we may also replace the non-zero entries of ∂_p with real-valued coefficients. We dedicate the rest of the paper to exploring the consequences of this fact.

2.2 A Parameterized Boundary Matrix Relaxation

Expressing the PBN via (2.15) enables us to exploit properties of the rank function which are advantageous in *parameterized* settings, i.e. settings where the input data is thought to be generated from a parameterized family. One such property is permutation invariance: given any $A \in \mathbb{R}^{n \times n}$, it is well known that $\mathrm{rank}(A) = \mathrm{rank}(P^T A P)$ for any permutation matrix P. Though the boundary matrices in (2.15) are given in filtration order to elucidate their structure, the permutation invariance of the rank function suggests they need not ordered at all to be evaluated—so long as the constitutive terms have the same non-zero pattern as their filtration-ordered counterparts, their ranks will be identical. In what follows, we re-define the boundary matrix to exploit this permutation invariance.

All of the notation given thus far, such as the definitions of the PH groups (2.4) and the the PBN (2.12), have used integer indices $(i, j) \in \Delta_+^N$ to describe the PH groups over a filtration pair $(K_\bullet, f)$ of size $|K| = N$. Equivalently, we have thus far implicitly assumed the range of the filter function $f : K \to I$ to be the typical index set $I = [N]$. In practice, it is more informative to interpret the persistence of a persistent-pair $(\sigma_i, \sigma_j) \in \mathrm{dgm}(K_\bullet)$ as $f(\sigma_j) - f(\tau_i)$, rather than as $j - i$, as the filter function f is often derived from geometrical settings and thus real-valued. In the spirit of (2.10), unless it is clear from the context, we alter our notation by re-defining $\beta_p^{i,j}$ using pairs $(i, j) \in \Delta_+$ from the upper-half plane $\Delta_+ = \{\, (x, y) \in \mathbb{R}^2 : y > x \,\}$ for the remainder of the paper.

Suppose that instead of being given a fixed pair $(K_\bullet, f)$, the filter function was parameterized $f : \mathcal{H} \times K \to \mathbb{R}$ with respect to some set $\mathcal{H}$. We give several application contexts where this kind of formulation occurs naturally in section 6. Consider the following reformulation of the boundary matrix ∂_p.

Definition 1 (Parameterized boundary matrix)**.** *Let K denote an abstract simplicial complex of size $|K| = m$, equipped with parameterized filtering function $f : K \times \mathcal{H} \to \mathbb{R}$. Assume K is ordered along a fixed but arbitrary linear extension $(K, \preceq^*)$ of the face poset of K. For fixed $(i, j) \in \Delta_+$, define the $\mathcal{H}$-*parameterized p-th boundary matrix $\hat{\partial}_p^{i,j}(h)$ at scale (i, j) *to be the $m \times m$ matrix ordered by $\preceq^*$ for all $h \in \mathcal{H}$, and whose entries (k, l) satisfy:*

$$\hat{\partial}_p^{i,j}(h)[k, l] = \begin{cases} \pm\, (S_i \circ f_h)(\sigma_k) \cdot (\tilde{S}_j \circ f_h)(\sigma_l) & \text{if } \sigma_k \in \partial_p(\sigma_l) \\ 0 & \text{otherwise} \end{cases} \tag{2.17}$$

where $S_i : \mathbb{R} \to \{0, 1\}$ is a step *function satisfying $S_i(x) = 0$ if $x \leq i$ and 1 otherwise, $\tilde{S}_i = 1 - S_i$, and $f_h(\sigma) = f(\sigma, h)$.*

We now show definition 1 simplifies the expression of the PBN in parameterized settings. To simplify the notation, we write $A^x = A^{*,x}$ for the setting where only columns up to x of A are being selected, and let $q = p + 1$. Let $V = \{v_1, v_2, \ldots, v_n\}$ denote a fixed vertex set. The parameterized PBN can be written as:

$$\begin{aligned} &\beta_p^{i,j} : \mathcal{H} \times \mathcal{P}(V) \to \mathbb{N} \\ &\qquad h, K \mapsto |K_i^p(h)| - \mathrm{rank}\big(\, \hat{\partial}_p^i(h) \,\big) - \mathrm{rank}\big(\, \hat{\partial}_q^j(h) \,\big) + \mathrm{rank}\big(\, \hat{\partial}_q^{i+\delta,j}(h) \,\big) \end{aligned} \tag{2.18}$$

where $\epsilon > 0$ is an arbitrarily small positive number. Observe (2.18) is essentially the same form as (2.15). By Proposition 2, we also have a parameterized multiplicity function for any rectangle $R = [i, j] \times [k, l]$ in the upper half-plane Δ_+ satisfying $i < j \leq k < l$:

$$\begin{aligned} &\mu_p^R : \mathcal{H} \times \mathcal{P}(V) \to \mathbb{N} \\ &\qquad h, K \mapsto \mathrm{rank}\big(\, \hat{\partial}_q^{j+\delta,k}(h) \,\big) - \mathrm{rank}\big(\, \hat{\partial}_q^{i+\delta,k}(h) \,\big) - \mathrm{rank}\big(\, \hat{\partial}_q^{j+\delta,l}(h) \,\big) + \mathrm{rank}\big(\, \hat{\partial}_q^{i+\delta,l}(h) \,\big) \end{aligned} \tag{2.19}$$

We give proofs the equivalence between these two expressions in the appendix. In summary, by restricting ourselves to homology defined over real-valued coefficients, we may parameterize the non-zero entries of the p-th boundary matrix ∂_p from 2.11 directly using $f : K \times \mathcal{H} \to \mathbb{R}$. Moreover, since $\mathrm{rank}(A) = \mathrm{rank}(P^T A P)$ for any permutation matrix P, it is clear that expressions of the counting invariants from Propositions 1 and 2 yield equivalent values to the parameterized expressions in (2.18) and (2.19).

3 Spectral PBN relaxation and its implications

One disadvantage in working with counting-type functions restricted to portions of real-plane is that they are integer-valued and thus not smooth. Namely, if (K, f_h) is an $\mathcal{H}$-parameterized filtration where K is a fixed simplicial complex and $f_h : K \to \mathbb{R}$ is a filter function, one may easily construct examples where $\|\mu_h^R - \mu_{h+\delta}^R\| \sim O(|K_p|)$ for some arbitrarily small perturbation $\delta > 0$. Intuitively, since both counting invariants are 0 outside of the portions of Δ_+ they restrict too, it's always possible to encounter such situations where small changes in the input affect the corresponding invariant in a non-Lipshitz way. In this section, we introduce a smooth relaxation to both invariants to counter this instability.

We begin with a spectral characterization of the rank function. Given a matrix $X \in \mathbb{R}^{n\times m}$ and its singular value decomposition (SVD) $X = U\Sigma V^T$, define the *rank* of X as the composition:

$$\mathrm{rank}(X) = \sum_{i=1}^{n} \mathrm{sgn}_+(\sigma_i(X)), \qquad \mathrm{sgn}_+(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases} \tag{3.1}$$

where $\Sigma = \mathrm{diag}(\{\sigma_1, \sigma_2, \ldots, \sigma_n\})$ are the singular values and $\mathrm{sgn}_+ : \mathbb{R} \to \{0, 1\}$ is the one-sided sign function. Like the betti numbers, the unstable nature of the rank function is due to its inherently combinatorial properties. In particular, the discontinuity in (3.1) manifests due to the one-sided sign function. The idea is, given a positive approximation parameter $\epsilon > 0$, we approximate the rank via:

$$(\,\forall\, x > 0\,) \qquad \mathrm{rank}(X) \approx \sum_{i=1}^{n} \phi(\sigma_i(X), \epsilon), \qquad \lim_{\epsilon\to 0^+} \phi(x, \epsilon) = \mathrm{sgn}_+(x) \tag{3.2}$$

where $\phi : \mathbb{R}_+ \times \mathbb{R}_{++} \to \mathbb{R}_+$ is an ϵ-parameterized family of continuous sgn approximation functions. Following the seminal work done by Chen et al. [7], Bi et al. [3] give a framework for constructing such functions, each of which when composed with the singular value function yield a continuous function that approximates the rank function to any pre-prescribed accuracy. The form of these ϕ functions is obtained by integration of smoothed variations $\hat{\delta}$ of the Dirac delta function δ:

$$\phi(x, \epsilon) := \int_{-\infty}^{x} \hat{\delta}(z, \epsilon)dz, \quad \forall\, z \geq 0, \epsilon > 0 \tag{3.3}$$

where $\hat{\delta}(z, \epsilon) = \nu(\epsilon)^{-1} p\big(z\nu \cdot (\epsilon)^{-1}\big)$ for some continuous density function $p : \mathbb{R}_+ \to \mathbb{R}_+$ and some choice of continuous increasing $\nu : \mathbb{R}_+ \to \mathbb{R}_+$ satisfying $\nu(0) = 0$ and $\nu(\epsilon) > 0$. In contrast to the rank function, if p is continuous on $\mathbb{R}_+$, then $\phi(\cdot, \epsilon)$ is continuously differentiable in $\mathbb{R}_+$, and if p is bounded above on $\mathbb{R}_+$, then $\phi(\cdot, \epsilon)$ is globally Lipshitz continuous on $\mathbb{R}_+$. Moreover, consider the operator Φ_ϵ defined as:

$$\begin{aligned} \Phi_\epsilon : \quad \mathbb{R}^{n\times m} &\to \mathbb{R}^{n\times m} \\ X &\mapsto U\,\mathrm{diag}(\tilde{\phi}(\sigma_1, \epsilon), \tilde{\phi}(\sigma_2, \epsilon), \ldots, \tilde{\phi}(\sigma_n, \epsilon))\, V^T \end{aligned} \tag{3.4}$$

where $\tilde{\phi}(x, \epsilon) = \phi(|x|, \epsilon)$ and $\epsilon > 0$ is fixed. Since $\phi(0, 0) = 0$, Φ_ϵ is a continuously differentiable operator in $\mathbb{R}^{n\times m}$ and one can verify that the nuclear norm of the operator is given by composing $\phi(\cdot, \epsilon)$ with the singular value function. We summarize this with a definition:

Definition 2 (Continuous Rank Approximation). *Given $X \in \mathbb{R}^{n\times m}$, a fixed approximation parameter $\epsilon > 0$, and a choice of $\phi : \mathbb{R}_+ \times \mathbb{R}_{++}$ satisfying* (3.3)*, define the* continuous rank approximation *$\|\Phi_\epsilon(X)\|_*$ of X as:*

$$\|\Phi_\epsilon(X)\|_* = \sum_{i=1}^{n} \phi(\sigma_i, \epsilon) \tag{3.5}$$

Apart from serving as a smooth approximation of the rank function, this operator turns out to have a variety of attractive properties related to monotonicity and differentiability. We summarize a few such properties below.

Proposition 3 ([3]). *The operator $\Phi_\epsilon : \mathbb{R}^{n\times m} \to \mathbb{R}^{n\times m}$ defined by* (3.4) *satisfies:*

1. *For any $\epsilon \leq \epsilon'$, $\|\Phi_\epsilon(X)\|_* \geq \|\Phi_{\epsilon'}(X)\|_*$ for all $X \in \mathbb{R}^{n\times m}$.*
2. *For any given $X \in \mathbb{R}^{n\times m}$ with rank $r = \mathrm{rank}(X)$, if ϵ satisfies $0 < \epsilon \leq \sigma_r/r$, then:*

   $$0 \leq r - \|\Phi_\epsilon(X)\|_* \leq c(r)$$

   where $c(r)$ is a positive constant that depends only on r.

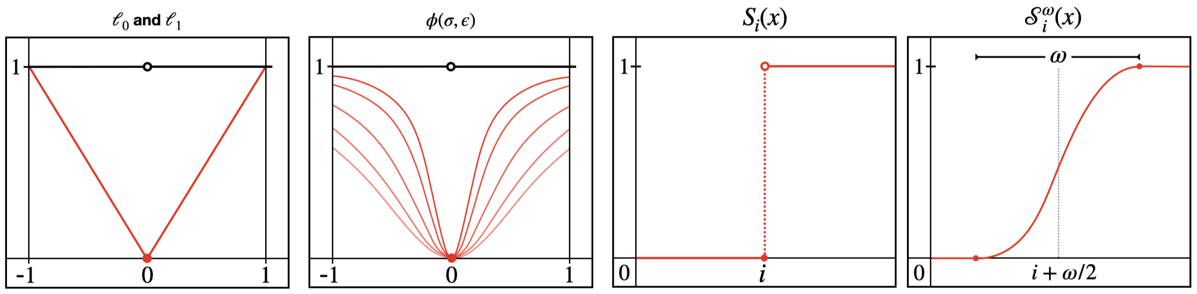

Figure 2: From left to right: the ℓ_1 norm (red) forms a convex envelope over the ℓ_0 (black) pseudo-norm on the interval $[-1, 1]$; $\tilde{\phi}(\cdot, \epsilon)$ at various values of ϵ, with $p(x) = 2x(x^2+1)^{-2}$ and $\nu(\epsilon) = \sqrt{\epsilon}$ (red) and at $\epsilon = 0$ (black); the step function $S_i(x)$ from definition 1; the smoothstep relaxation $\mathcal{S}_i^\omega$ from (3.6).

3. *The function $\|\Phi_\epsilon(X)\|_*$ is globally Lipshitz continuous and semismooth on $\mathbb{R}^{n \times m}$.*

It's worth noting that $\|\Phi_\epsilon(X)\|_*$ is not necessarily differentiable on $\mathbb{R}^{n\times m}$, but it is differentiable on the positive semi-definite cone $\mathbb{S}_+^n$, and that the notion of semismoothness here refers to the existence certain directional derivatives in the limit as $\epsilon \to 0^+$, see [] for more details.

Smoothstep: To adapt these relaxations to the counting functions defined in equations (2.8) and (2.16), we need to modify the expression of the boundary chains (2.3) to vary continuously in $\mathcal{H}$. Fortunately, this requires a simple augmentation to the step function from the one in definition 1. In particular, we swap out the step functions $S : \mathbb{R} \to \{0, 1\}$ from (2.17) with continuous *smoothstep* functions $\mathcal{S} : \mathbb{R} \to [0, 1]$. These are clamped "S-curve" functions which interpolate the discontinuous step portion of S along fixed bounds $(a, a + \omega)$, which are given by:

$$\mathcal{S}_a^\omega(x) = \begin{cases} 0 & x \leq a \\ S_n\big(\omega^{-1}((a+\omega) - x)\big) & a < x < a + \omega \\ 1 & a + \omega \leq x \end{cases} \tag{3.6}$$

where $S_n : [0, 1] \to [0, 1]$ is a generic n-th order polynomial whose coefficients are fixed such that $S_n(0) = 0$ and $S_n(1) = 1$. These polynomials have found applications computer graphics and machine learning applications []. Our motivation to use them here is based on the observations shown in Figure 2: by substituting $\mathcal{S}_n^\omega$ for the step functions in (2.17), we have a version of the parameterized boundary matrix whose entries varies continuously in $\mathcal{H}$ but which retain the same rank as their discontinuous counterparts. Moreover, by composing with (2.16) and substituting $\mathrm{rank}(\cdot)$ with $\Phi(\cdot, \epsilon)$ for some $\epsilon > 0$, we obtain ϵ-approximate, continuously-varying multiplicity function.

Definition 3 (Smooth Multiplicity)**.** *Let K denote a fixed simplicial complex and $f : K \times \mathcal{H} \to \mathbb{R}$ a parameterized family of filter functions that varies continuously in $\mathcal{H}$. Define the* continuous multiplicity function *over some fixed $R = [i, j] \times [k, l]$ as:*

$$\hat{\mu}_{p,\epsilon}^R(h) = \|\Phi_\epsilon^h(\hat{\partial}_{p+1}^{j+\delta,k})\|_* - \|\Phi_\epsilon^h(\hat{\partial}_{p+1}^{i+\delta,k})\|_* - \|\Phi_\epsilon^h(\hat{\partial}_{p+1}^{j+\delta,l})\|_* + \|\Phi_\epsilon^h(\hat{\partial}_{p+1}^{i+\delta,l})\|_* \tag{3.7}$$

where $\Phi_\epsilon^h(\partial) = (\Phi_\epsilon \circ \partial)(h)$ for any $h \in \mathcal{H}$, and $\tilde{\partial}_p$ is $\mathcal{H}$-parameterized p-th boundary matrix from 1 with step functions S_* replaced by the smoothstep $\mathcal{S}_*^\omega$ from (3.6)*, for some fixed $\omega > 0$.*

Since the sum $f + g$ of two Lipshitz functions f and g is also Lipshitz, it is easy to verify that $\hat{\mu}_p^R$ is Lipshitz continuous by combining the smoothness of $\mathcal{S}_*^\omega$ with the global Lipshitz continuity of (3.5). We omit the definition of the smooth PBN relaxation $\hat{\beta}_p^{i,j}$, as its form is synonymous with definition 3.

3.1 Basic properties

Basic properties of $\hat{\mu}$.

3.2 Stability

Stability results

4 Computing

In this section we discuss the computational details involved in evaluating the rank function on the boundary matrices whose columns take the form (2.3). As (3.1) is defined completely in terms of the singular values of X, and the singular values of X are given by the square roots of eigenvalues of XX^T (or X^TX), we focus on iterative methods specialized for symmetric matrices.

4.1 The Lanczos iteration

For a real, square matrix A of order n, the quadratic form x^TAx defines a continuous real-valued function of $x \in \mathbb{R}^n$. When A is symmetric positive definite, the implicit equation $x^TAx = 1$ defines an n-dimensional ellipsoid $y^T\Lambda y = 1$ whose n principle axes are eigenvectors $\{v_i\}_{i=1}^n$ and whose lengths are the squares of eigenvalues $\Lambda(A)$ of A. Each eigen-pair (λ, v) satisfies $Av = \lambda v$, and when A is symmetric, every λ is real-valued and every pair of eigenvectors $v, u \in \mathbb{R}^n$ whose corresponding eigenvalues $\lambda \neq \lambda'$ are orthogonal. Thus, we may reveal the spectrum $\Lambda(A)$ of A—effectively the lengths of the aforementioned ellipsoid—via orthogonal diagonalization:

$$A = V\Lambda V^T = \sum_{i=1}^{n} \lambda_i v_i v_i^T \tag{4.1}$$

Factorizing A as in (4.1) is known as the *symmetric eigenvalue problem.* Computing eigen decompositions of symmetric matrices generally consists of two phases: (1) reduction to tridiagonal form $Q^TAQ = T$ via orthogonal similarity transformations $Q = Q_1Q_2 \dots Q_{n-2}$, and (2) diagonalization of the tridiagonal form $T = Y\Theta Y^T$. Note the latter may be performed in $O(n \log n)$ time [15], whereas the former is effectively bounded below by $\Omega(n^3)$ for dense full-rank matrices using non-Strassen-like operations, and thus this reduction to tridiagonal form dominates the computation. Lanczos [16] proposed the *method of minimized iterations*—now known as the *Lanczos method*—as an attractive alternative for reducing A into a tridiagonal form and thus revealing its spectrum. As the Lanczos method is the central iterative method we study in this effort, we review it below.

The means by which the Lanczos method estimates eigenvalues is by projecting onto successive Krylov subspaces. Given a large, sparse, symmetric $n \times n$ matrix A with eigenvalues $\lambda_1 \geq \lambda_2 > \dots \geq \lambda_r > 0$ and a vector $v \neq 0$, the order-j Krylov subspaces of the pair (A, v) are the spaces spanned by:

$$\mathcal{K}_j(A, v) := \operatorname{span}\{v, Av, A^2v, \dots, A^{j-1}\} = \operatorname{range}(K_j(A, v)) \tag{4.2}$$

where $K_j(A, v) = [v \mid Av \mid A^2v \mid \cdots \mid A^{j-1}]$ are their corresponding Krylov matrices. Krylov subspaces arise naturally from using the minimal polynomial of A to express A^{-1} in terms of powers of A. In particular, if A is nonsingular and its minimal polynomial has degree m, then $A^{-1}v \in K_m(A, v)$ and $K_m(A, v)$ is an invariant subspace[1] of A. Since A is symmetric, the spectral theorem implies that A is orthogonally diagonalizable and that we may obtain $\Lambda(A)$ by generating an orthonormal basis for $\mathcal{K}_n(A, v)$. To do this, the Lanczos method constructs successive QR factorizations of $K_j(A, v) = Q_jR_j$ for each $j = 1, 2, \dots, n$. Due to A's symmetry and the orthogonality of Q_j, we have $q_k^TAq_l = q_l^TA^Tq_k = 0$ for $k > l + 1$, implying the corresponding $T_j = Q_j^TAQ_j$ have a tridiagonal structure:

$$T_j = \begin{bmatrix} \alpha_1 & \beta_1 & & & \\ \beta_1 & \alpha_2 & \beta_2 & & \\ & \beta_2 & \alpha_3 & \ddots & \\ & & \ddots & \ddots & \beta_{j-1} \\ & & & \beta_{j-1} & \alpha_j \end{bmatrix}, \ \beta_j > 0, \ j = 1, 2, \dots, n \tag{4.3}$$

Unfortunately, unlike the spectral decomposition $A = V\Lambda V^T$—which identifies a diagonalizable A with its spectrum $\Lambda(A)$ up to a change of basis $A \mapsto M^{-1}AM$—there is no canonical choice of T_j due to the arbitrary choice of v. However, there is a connection between the iterates $K_j(A, v)$ and the full tridiagonalization of A: if $Q^TAQ = T$ is tridiagonal and $Q = [\, q_1 \mid q_2 \mid \cdots \mid q_n \,]$ is an $n \times n$ orthogonal matrix $QQ^T = I_n = [e_1, e_2, \dots, e_n]$, then:

$$K_n(A, q_1) = QQ^TK_n(A, q_1) = Q[\, e_1 \mid Te_1 \mid T^2e_1 \mid \cdots \mid T^{n-1}e_1 \,] \tag{4.4}$$

is the QR factorization of $K_n(A, q_1)$. Thus, tridiagonalizing A with respect to a unit-norm q_1 determines Q. Indeed, the Implicit Q Theorem [14] asserts that if an upper Hessenburg matrix $T \in \mathbb{R}^{n\times n}$ has only positive elements on

[1]Recall that if $S \subseteq \mathbb{R}^n$, then S is called an *invariant subspace* of A or *A-invariant* iff $x \in A \implies Ax \in S$ for all $x \in S$.

its first subdiagonal and there exists an orthogonal matrix Q such that $Q^T AQ = T$, then Q and T are *uniquely* determined by (A, q_1). As a result, given an initial pair (A, q_1) satisfying $\|q_1\| = 1$, we may restrict and project A to its j-th Krylov subspace T_j via:

$$AQ_j = Q_j T_j + \beta_j q_{j+1} e_j^T \qquad (\beta_j > 0) \tag{4.5}$$

where $Q_j = [\, q_1 \mid q_2 \mid \cdots \mid q_j \,]$ is an orthonormal set of vectors mutually orthogonal to q_{j+1}. Equating the j-th columns on each side of (4.5) and rearranging the terms yields the *three-term recurrence*:

$$\beta_j \, q_{j+1} = Aq_j - \alpha_j \, q_j - \beta_{j\text{-}1} \, q_{j\text{-}1} \tag{4.6}$$

where $\alpha_j = q_j^T Aq_j$, $\beta_j = \|r_j\|_2$, $r_j = (A - \alpha_j I)q_j - \beta_{j\text{-}1} q_j$, and $q_{j+1} = r_j/\beta_j$. Equation (4.6) is a variable-coefficient second-order linear difference equation, and it is a known fact that such equations have unique solutions: if $(q_{j\text{-}1}, \beta_j, q_j)$ are known, then $(\alpha_j, \, \beta_{j+1}, q_{j+1})$ are completely determined. The sequential process that iteratively builds T_j by via the recurrence from (4.6) is called the *Lanczos iteration.* Note that if A is singular and we encounter $\beta_j = 0$ for some $j < n$, then range$(Q_j) = \mathcal{K}_j(A, q_1)$ is an A-invariant subspace, the iteration stops, and we have solved the symmetric eigenvalue problem (4.1): $\Lambda(T_j) = \Lambda(A)$, $j = \text{rank}(A)$, and T_j is orthogonally similar to A.

The Lanczos iteration and its many variants are part of a family of so-called "matrix free" methods—obtaining an eigen-decomposition of a symmetric real matrix A requires only a matrix-vector $v \mapsto Av$ operator. Observe that since A is not modified at all during the computation, the entire iteration may be carried out without explicitly storing A in memory. In fact, the three-term recurrence from (4.6) implies the Lanczos iteration requires just three $O(n)$-sized vectors and a few $O(n)$ vector operations. We summarize these benefits with a Lemma.

Lemma 2 ([18, 19])**.** *Given a symmetric rank-r matrix $A \in \mathbb{R}^{n \times n}$ whose matrix-vector operator $A \mapsto Ax$ has complexity $O(\mathcal{M}(n))$ time, the Lanczos iteration computes $\Lambda(A) = \{\lambda_1, \lambda_2, \ldots, \lambda_r\}$ in $O(\max\{\mathcal{M}(n), n\} \cdot r)$ time and $O(n)$ storage complexity, when computation is done in exact arithmetic.*

As in [18], the assumption of exact arithmetic simplifies both the presentation of the theory and the corresponding complexity statements. Although this assumption is unrealistic in practical settings, it gives us a grounded expectation of what is possible to achieve with any *finite-precision* algorithm based on the Lanczos method.

Corollary 3. *Given the same inputs as Lemma 2, any implementation that computes $\Lambda(A) = \{\lambda_1, \lambda_2, \ldots, \lambda_r\}$ using the Lanczos iteration in finite-precision arithmetic requires $\Omega(\max\{\mathcal{M}(n), n\} \cdot r)$ time and $\Omega(n)$ storage complexity.*

In practice, finite-precision arithmetic introduces both rounding and cancellation errors into the computation which manifest as loss of orthogonality between the Lanczos vectors. These errors not only affect the iterations convergence towards an in invariant subspace, but actually muddle the termination condition entirely. Much effort has been expended—spanning several decades []—developing orthogonality-enforcement schemes that retain the simplicity of the Lanczos iteration without increasing either the time or storage complexities by non-trivial factors. Because these extensions are varied, non-trivial, and multifaceted, we defer their discussion to section A.1.

As Lemma 2 makes clear, the usefulness of the Lanczos iteration hinges on the availability of a fast matrix-vector product. In general, for any symmetric $A \in \mathbb{R}^{n \times n}$ rank-r matrix with an average of ν nonzeros per row, approximately $(2\nu + 8)n$ flops are needed for a single Lanczos step, implying a $O(n\nu r)$ time complexity for a single iteration [14]. For dense matrices, we recover the same $O(rn^2)$ complexity required by direct methods. In the next section, we demonstrate how to reduce this complexity using the structure of the boundary operator.

4.1.1 Combinatorial Laplacians

The efficiency of the Lanczos method depends on two crucial components: the three-term recurrence and fast matrix-vector product. The former only arises in the decomposition of symmetric matrices and the latter depends the existence of combinatorial structure in the $x \mapsto Ax$ operation. Though ∂ is not symmetric, it is well known that the non-zero part of the spectrum of both $\partial\partial^T$ and $\partial^T\partial$ contain the same information as the singular values of ∂ []. Indeed, since $L = \partial_1\partial_1^T$ is the well studied *graph Laplacian*, we can consider the study of spectra of *combinatorial Laplacians*—generalizations of the graph Laplacian for $p > 0$—as the study of singular values of boundary operators. In contrast, whereas $x \mapsto Lx$ can be carried out in essentially linear time time in the number of edges of the underlying graph, it is not immediately clear whether $x \mapsto \partial_p\partial_p^T x$ can be carried out in a time linear in the number of $p + 1$ simplices for more general $p > 1$. Towards revealing exploitable structure to accelerate the rank computation, we examine this more in detail.

Given a simple undirected graph $G = (V, E)$, let $A \in \mathbb{B}^{n \times n}$ denote its binary adjacency matrix satisfying $A[i, j] = 1$ if the vertices $v_i, v_j \in V$ are path-connected in G, and 0 otherwise. Moreover, denote with D the diagonal degree matrix $D = \text{diag}(\{ \deg(v_i) \})$, where $\deg(v_i) = \sum_{j \neq i} A[i,j]$. The *graph Laplacian* L, defined in terms of adjacency, incidence, and element-wise relations, is defined as:

$$L = D - A = \partial_1 \circ \partial_1^T \,, \qquad L\,[i,j] = \begin{cases} \deg(v_i) & \text{if } i = j \\ -1 & \text{if } i \sim j \\ 0 & \text{otherwise} \end{cases} \tag{4.7}$$

where we use the notation $i \sim j$ to denote vertices $v_i, v_j \in V$ which are path-connected in G. Its well known that L is symmetric, positive semi-definite, and has a combinatorial structure that captures the connectivity structure of G. Moreover, the linear and quadratic forms of L may be succinctly expressed using the above path-connected relation:

$$(\forall\, x \in \mathbb{R}^n\,) \qquad (Lx)_i = \deg(v_i) \cdot x_i - \sum_{i \sim j} x_j \,, \qquad x^T L x = \sum_{i \sim j} (x_i - x_j)^2 \tag{4.8}$$

Observe that, if $G = (V, E)$ has n vertices and m edges, the matrix-vector product in (4.8) can be evaluated in $O(m)$ time and $O(n)$ storage via two linear-time edge traversals: one to accumulate the (weighted) degree of each vertex, and one to subtract off components x_j from incident edges at i. We would like to generalize this procedure for simplicial complexes. For simplicial complexes, the generalization of the graph laplacian is p-th *combinatorial Laplacian* Δ_p, which is given by:

$$\Delta_p(K) = \underbrace{\partial_{p+1} \circ \partial_{p+1}^T}_{L_p^{\text{up}}} + \underbrace{\partial_p^T \circ \partial_p}_{L_p^{\text{dn}}} \tag{4.9}$$

Note when $p = 0$, we have $\Delta_0(K) = \partial_1 \partial_1^T = L$ since $L_0^{\text{dn}} = 0$. As the orientation of higher dimensional simplices changes the sign of the off-diagonal entries in L_p^*, it is not immediately clear whether (4.8) generalizes for $p > 0$. In particular, writing down the explicit expression of the matrix-vector product of L_p^{up} requires evaluating the sign of each oriented $[\tau] \in K^p$ in the boundary of each of its proper 1-d cofaces $[\sigma] \in K^{p+1}$. To make this clear, we require more notation.

Given a pair (K, f) where K is a simplicial complex of dimension $\dim(K) \geq p+1$ and $f : K \to \mathbb{R}_+$ is an arbitrary function equipped to the simplices of K, define the combinatorial p-th weighted up-Laplacian of the pair (K, f) by:

$$L_p^{\text{up}}(K, f) := W_p \circ \partial_{p+1} \circ W_{p+1}^2 \circ \partial_{p+1}^T \circ W_p$$

To characterize the explicit forms of L_p^{up} for $p \geq 1$, we need an appropriate generalization the path-connected relation from (4.8). To accomplish this, let $\text{co}(\tau) = \{ \sigma \in K^{p+1} \mid \tau \subset \sigma \}$ the set of proper cofaces of $\tau \in K^p$, or *cofacets*, and let $d_f(\tau) = \sum_{\sigma \in \text{co}(\tau)} f(\sigma)^2$ denote the *degree* of τ. Since K is a simplicial complex, note that if $\text{co}(\tau) \cap \text{co}(\tau') \neq \emptyset$, then the τ, τ' not only share a cofacet σ, but this cofacet $\{\sigma\} = \text{co}(\tau) \cap \text{co}(\tau')$ is *unique* []. Thus, given two distinct p-simplices $\tau \neq \tau'$ in K^p, we use the notation $\tau \overset{\sigma}{\sim} \tau'$ to denote this cofacet, and we write:

$$L_p^{\text{up}}[i,j] = \begin{cases} d_f(\sigma) \cdot f(\tau_i)^2 & \text{if } i = j \\ (-1)^{s_{ij}} f(\tau_i) f(\tau_j) f(\sigma)^2 & \text{if } \tau_i \overset{\sigma}{\sim} \tau_j \\ 0 & \text{otherwise} \end{cases} \tag{4.10}$$

where $s_{ij} = \text{sgn}([\tau_i], \partial[\sigma]) \cdot \text{sgn}([\tau_j], \partial[\sigma])$. In general, the sign of the coefficients in $[\tau]$ and $[\tau']$ depends on the position as summands in $\partial[\sigma]$, which itself depends on the orientation of $[\sigma]$. However, observe that regardless of the orientation σ takes, the sign of the product s_{ij} is invariant. Moreover, since we assume the poset order on K is induced from the vertex order, the sign pattern each $[\sigma]$ takes induced by $(V, \preceq)$ is constant, thus we may access s_{ij} in $O(1)$ time.

Consider the operation $x \mapsto L_p^{\text{up}} x$ for any $x \in \mathbb{R}^n$, where $n = |K^p|$. Let $m = |K^{p+1}|$, and as before assuming the simplices of K are ordered by some fixed linear extension $(V, \preceq)$ of the vertex set. Observe that, for any pair $\tau, \tau' \in K^p$ satisfying $\tau \overset{\sigma}{\sim} \tau$ with proper coface σ, evaluating any individual term from (4.10) in $O(1)$ requires knowing the positions $i, j \in [n]$ of the p-simplices in total order $K^p = (\tau_1, \tau_2, \ldots, \tau_n)$. Using the theory of universal hashing functions, we could build a perfect hash table $h : K^p \to [n]$ which requires $O(1)$ average time complexity, but this would still exhibit $O(n)$ worst case. Alternatively, we could also use binary search to reduce this to $O(\log n)$. Instead, we take advantage of the fact that again the order on K^p is fixed by building a *perfect minimal hash function*.

Since a p-simplex has $p+1$ proper faces in its boundary, and we need to iterate through the simplices $\sigma \in K^{p+1}$ at most a constant number of times, we may combine the above observations into a lemma.

Lemma 3. *For any $p \geq 0$ and simplicial pair (K, f), there exists a two-phase algorithm for computing the product:*

$$x \mapsto L_p^{up} x = (W_p \circ \partial_{p+1} \circ W_{p+1}^2 \circ \partial_{p+1}^T \circ W_p) x \tag{4.11}$$

in $O(m(p+1))$ time, $O(n)$ storage.

From a practical perspective, $p \geq 0$ is generally a very small constant—typically no more than 2—thus we conclude that we can evaluate $x \mapsto \partial_{p+1} \partial_{p+1}^T x$ is $\approx O(m)$ time, essentially linear in the number of $p+1$ simplices. More details on this, along with pseudocode of the two-phase algorithm for general $p \geq 0$, is given in appendix C.

4.1.2 Convergence rate

The ability of the Krylov subspace iteration to capture the extremal portions of the spectrum remains unparalleled, and by using $O(n)$ memory, the Lanczos iteration uses optimal memory. As mentioned in section **??**, when the computation is carried out in finite-precision arithmetic, one may observe loss of orthogonality in the Lanczos vectors. Fortunately, the connection between the Lanczos method and the Rayleigh quotient ensures *eventual* termination of the procedure under by restarting the Lanczos method, and continue with the iteration until the spectrum has been approximated to some prescribed accuracy. Unfortunately, if the number of iterations k is e.g. larger than n^2, then the method may approach to $O(r \max(\mathcal{M}(n), n), n) \approx O(n^3)$ complexity one starts with. If the supplied matrix-vector product operation is fast, the number of iterations k needed for convergence of the Lanczos method becomes the main bottleneck estimating the spectrum of A.

Loss of orthogonality can be mitigated by re-orthogonalizaing against all previous Lanczos vectors, but this increases the Lanczos complexity to $\approx O(n^2)$ per iteration. Thus, the goal is strike a balance: find a way to keep all n Lanczos numerically orthonormal, so as to ensure super-linear convergence of the Ritz values θ, but do so using $c \cdot n$ memory, where c is a relatively small constant.

Since rates of convergence α increases the number of correct digits by an expoentnial rate with factor *alpha*, any super-linear convergent ($\alpha > 1$) method needs at most c terms to approximate an eigen-pair up to numerical precision. In the context of the Lanczos method, achieving even quadratic convergence would imply the number of iterations needed to obtain machine-precision is bounded by $T(c \cdot \mathcal{M}(n) \cdot r)$, where c is a small constant. We say that a method which achieves *superlinear* convergence has complexity *essentially* $O(c \cdot n) \approx O(n)$.

Among the more powerful methods for achieving super linear convergence towards a given eigenvalue λ is the Jacobi-Davidson method. This method seeks to correct:

Solving for t results in the *correction equation*

$$(I - uu^T)(A - \sigma I)(I - uu^T)t = \theta u - Au \tag{4.12}$$

where, since u is unit-norm, $I - uu^T$ is a projector onto the complement of span(u). It's been shown that solving exactly for this correction term essentially constructs an cubically-convergent sequence towards some $\theta \mapsto \lambda$ in the vicinity of σ. Solving for the correction equation exactly is too expensive, sparking efforts to approximate it. It turns out that, just as the Lanczos method in exact arithmetic is highly related to the conjugate gradient method for solving linear systems, solving for the correction equation exactly is in some ways conceptually similar to making an Newton step in the famous Newtons method from nonlinear optimization. Since (**??**) is approximated, the JD method is often called in the literature akin to making an "inexact newton step" [].

The JD method with inexact Newton steps yields an individual eigenvalue estimate with quadratic convergence—*essentially* $O(m)$ time after some constant number matrix-vector products and $O(n)$ memory. The Lanczos method, in contrast, estimates all eigenvalues in essentially quadratic time if the convergence rate is superlinear. Pairing these two methods is a non-trivial endeavor. In a sequence of papers, Stathopoulos et al [] investigated various strategies for approximately solving the correction equation. In , they give both theoretical and empricial evidence to suggest that by employing generalized Davidson and Jacobi-Davidson like solvers within an overarching Lanczos paradigm, they achieve nearly optimal methods for estimating large portions of the spectrum using $O(1)$ number of basis vectors. By approximating the inner iterations with the symmetric Quasi-Minimal Residual (QMR) method, they argue that JD cannot converge more than three times slower than the optimal method, and empirically they find the constant factor to be less than 2.

4.2 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 [13]. 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 [13, 2, 11] for details.

Given a filtration (K, f) of size $m = |K|$ with filter $f : K \to [m]$, the reduction algorithm in form given in [13] computes $\mathrm{dgm}_p(K; \mathbb{Z}/2) = \{ (\tau_1, \sigma_1), (\tau_2, \sigma_2), \dots, (\tau_k, \sigma_k) \}$ 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 [11]), 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 [11]. For clique complexes, the apparent pairs optimization—which preemptively removes zero-persistence pairs from the computation prior to the reduction—has been empirically observed to reduce the number of columns needing reduced for clique complexes by $\approx 98 - 99\%$ [2]. Numerous other optimizations, including e.g. the *clearing optimization*, the use of *cohomology*, the *implicit reduction* technique, have further reduced both the non-asymptotic constant factors of the reduction algorithm significantly, see [2] and references therein for a full overview.

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

As noted in [11], 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.

5 Parameterized setting & Perturbation theory

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

Rayleigh Ritz values Though the Lanczos iterations may be used to obtain the full tridiagonalization $A = QTQ^T$, intermediate spectral information is readily available in T_j, for $j < \mathrm{rank}(A)$. Diagonalizing $T_j = Y \Theta Y^T$

yields value/vector pairs $\{(\theta_1^{(j)}, y_1^{(j)}), \dots, (\theta_j^{(j)}, y_j^{(j)})\}$ satisfying $w^T(Ay - \theta y) = 0$ for all $w \in \mathcal{K}_j(A, q_1)$, called *Ritz pairs.* The values θ are called *Ritz values* and their associated vectors $v = Qy$ in the range of Q are called *Ritz vectors.* From the Ritz perspective, the Lanczos iteration implicitly maintains two orthonormal basis for $K_j(A, q_1)$—a Lanczos basis Q and the Ritz basis Y:

$$A = QTQ^T = QY\Theta Y^T Q^T \iff AQY = QY\Theta$$

In principle, the Lanczos basis $\{q_i\}_{i=1}^j$ changes each iteration, while the Ritz basis $\{Qy_i^{(j)}\}_{i=1}^j$ changes after each subspace projection. The way in which the Ritz values approach the spectrum of A is well-studied [], as they are known to be Rayleigh-Ritz approximations of A's eigenpairs $\Lambda(A) = \{(\lambda_1, v_1), \dots, (\lambda_j, v_j)\}$, and they are collectively known to be optimal in the sense that $T_k = B$ is the matrix that minimizes $\|AQ_k - Q_k B\|_2$ over the space of all $k \times k$ matrices. Moreover, Ritz values contain intrinsic information of the distance between $\Lambda(T_j)$ and $\Lambda(A)$. To see this, note that:

$$\|Av_i^{(j)} - v_i^{(j)}\theta_i^{(j)}\| = \beta_i^{(j)} = \beta_{j+1} \cdot |\langle e_j, y_i^{(j)} \rangle| \tag{5.1}$$

Thus, we need not necessarily keep the Lanczos vectors Q in memory to monitor how close the spectra of the T_j's approximate $\Lambda(A)$. In fact, it is known that the Ritz values $\{\theta_1^{(1)}, \theta_1^{(2)}, \dots, \theta_1^{(j)}\}$ of T_j satisfy:

$$|\lambda - \theta_i^{(j)}| \leq \big(\beta_i^{(j)}\big)^2 / \big(\min_{\mu} |\mu - \theta_i^{(j)}|\big) \tag{5.2}$$

The full convergence of the Ritz values to the eigenvalues of A is known to converge at a rate that depends on the ratio between λ_1/λ_n. A full analysis is done in terms of Chebychev Polyonomials in [14]. In practice, it has been observed that the Lanczos iteration converges super-linearly towards the extremal eigenvalues of the spectrum, whereas for interior eigenvalues one typically must apply a shifting scheme.

5.1 (1-ϵ) Approximations

Note that when $\nu(\epsilon) = \sqrt{\epsilon}$ and $p(x) = 2x(x^2+1)^{-2}$, Equation (3.5) reduces to:

$$\|\Phi_\epsilon(X)\|_* = \sum_{i=1}^n \frac{\sigma_i(X)^2}{\sigma_i(X)^2 + \epsilon} = \mathrm{Tr}\left[X^T(XX^T + \epsilon I_n)X\right]$$

which is the generic rank approximation studied by []. Moreover, by the cyclic property of the trace operator, since $X = \partial_1$ here and $L = \partial_1\partial_1^T = [l_1, l_2, \dots, l_n]$ we have:

$$\|\Phi_\epsilon(X)\|_* = \mathrm{Tr}[(L + \epsilon I_n)^{-1}L] = \sum_{i=1}^n (L_\epsilon^{-1} l_i)_i$$

Thus we may solve this problem by solving n sparse linear systems of that take the form $Ax = b$, where here A is a Laplacian matrix with an $\epsilon \cdot I_n$ addition to it's diagonal. From this expression, it's clear that ϵ not only plays the role of an accuracy parameter, but as a smoothing parameter. Indeed, define the condition number

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

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

where M is symmetric positive definite.

$$\min_{x \perp \mathbf{1}} \frac{1}{2} x^T(L + \epsilon I_n)x - b^T x \tag{5.3}$$

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

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

6 Applications

As topological invariants, Betti numbers are invariant under homeomorphisms: any pair of filtrations (K, f) and (K', f') that are homotopy equivalent have identical homology classes and thus isomorphic persistence diagrams. This invariance can be a useful thing at the level of homology, as non-homeomorphic spaces can sometimes be differentiated by inspecting differences between their corresponding homology classes. However, invariance under homeomorphisms can at times discard geometric information that may be useful for differentiating objects. For example, consider creating a classifier for the alphabet of English characters in the font shown below:

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

If one were to triangulate images of each of the letters shown above and compute their Betti numbers, one would find just three homology classes: one class for those letters that have two holes (B), one class of letters that have one hole (A, D, O, P, Q, and R), and one class for the rest of the letters, which collapse to points. It would be beneficial to have an invariant that was sensitive to the geometries between shapes, but also also stable in some sense.

Directional Transform

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

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

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

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

$$K_\bullet = K(v)_\alpha = \{\, x \in X \mid \langle x, v \rangle \leq \alpha \,\} \tag{6.1}$$

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

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

$$\begin{aligned} \mathrm{PHT}(K) : S^{d-1} &\to \mathcal{D}^d \\ v &\mapsto \big(\, \mathrm{dgm}_0(K, v), \mathrm{dgm}_1(K, v), \dots, \mathrm{dgm}_{d-1}(K, v) \,\big) \end{aligned} \tag{6.2}$$

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

The primary result of [] is that the PHT is injective on the space of subsets of R^d that can be written as finite simplicial complexes[2], which we denote as $\mathcal{K}_d$. Equivalently, $\mathcal{K}_d$ decomposes space of all pairs (K, f) under the equivalence $(K, f) \sim (K, f')$ when $f(K) = f'(K)$.

A Appendix

A.1 Finite-precision arithmetic

It is well established in the literature that the Lanczos iteration, as given in its original form, it effectively useless in practice due to significant rounding and cancellation errors. Such errors manifest as loss of orthogonality between the computed Lanczos vectors, which drastically affects the convergence of the method. At first glance, this seems to be a simple numerical issue, however the analysis from Parlett [18] showed, loss of orthogonality is not merely the result of gradual accumulation of roundoff error—it is in fact is intricately connected to the convergence behavior of Lanczos iteration. One obvious remedy to this is to reorthogonalize the current Lanczos vectors $\{q_{j-1}, q_j, q_{j+1}\}$ against all previous vectors using Householder matrices [14]—a the *complete reorthogonalization* scheme. This process guarantees orthogonality to working precision, but incurs a cost of $O(jn)$ for each Lanczos step, effectively placing the iteration back into the cubic time and quadratic memory regimes the direct methods exhibit. A variety of orthogonality enforcement schemes have been introduced over years, including implicit restart schemes, selective reorthogonalization, thick restarts, block methods, and so on; see [] for an overview.

A.2 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.

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

Example A.1 (Adapted from [17])**.** Suppose the vertices of G are ordered and labeled from 1 to n arbitrarily 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. Given such a set $X \subseteq V$, let $X' = V \setminus X$ denote its complement set. By L's definition, 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 [17].

Applications

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[3]
2. For fixed $x, x' \in X$, $d_X(\cdot)(x, x') : \mathbb{R} \to \mathbb{R}_+$ is continuous.

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

A.3 Proofs

Proof of rank equivalence

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

Proof of Lemma 1

Proof. The Pairing Uniqueness Lemma [11] 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 $\mathrm{low}_R[j] = i$ if and only if $r_\partial(i, j) = 1$. As a result, for $1 \leq i < j \leq m$, we have:

$$\mathrm{low}_R[j] = i \iff r_R(i, j) \neq 0 \iff r_\partial(i, j) \neq 0 \tag{A.1}$$

Extending this result to equation (2.14) 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 (2.12) 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:

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

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

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 $\{ \mathrm{col}_{R_{p+1}[k]} \neq 0 \mid k \in [j] \} \in \mathrm{Im}(\partial_{p+1}^{1,j})$. Now, since the low entries of the non-zero columns of R_{p+1} are unique, we have:

$$\dim(Z_p(K_i) \cap B_p(K_i)) = |\Gamma_p^{i,j}| \tag{A.2}$$

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

$$|\bar{\Gamma}_p^{i,j}| = \dim(B_p(K_j)) - |\Gamma_p^{i,j}| = \mathrm{rank}(R_{p+1}^{i+1,j}) \tag{A.3}$$

Combining equations (A.2) and (A.3) yields:

$$\dim(Z_p(K_i) \cap B_p(K_j)) = |\Gamma_p^{i,j}| = \dim(B_p(K_j)) - |\bar{\Gamma}_p^{i,j}| = \mathrm{rank}(R_{p+1}^{1,j}) - \mathrm{rank}(R_{p+1}^{i+1,j}) \tag{A.4}$$

Observing the final matrices in (A.4) are *lower-left* submatrices of R_{p+1}, the final expression (2.15) follows by applying Lemma 1 repeatedly. □

Proof of boundary matrix properties

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

□

References

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

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

[3] 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.

[4] 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.

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

[6] 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.

[7] Chunhui Chen and Olvi L Mangasarian. A class of smoothing functions for nonlinear and mixed complementarity problems. *Computational Optimization and Applications*, 5(2):97–138, 1996.

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

[9] 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.

[10] 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.

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

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

[13] 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.

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

[15] Ming Gu and Stanley C Eisenstat. A divide-and-conquer algorithm for the symmetric tridiagonal eigenproblem. *SIAM Journal on Matrix Analysis and Applications*, 16(1):172–191, 1995.

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

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

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

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

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

A Boundary matrix factorization

Definition 4 (Boundary matrix decomposition)**.** *Given a filtration $K_\bullet$ with m simplices, let ∂ denote its $m \times m$ filtered boundary matrix. We call the factorization $R = \partial V$ the* boundary matrix decomposition *of ∂ if:*

I1. V is full-rank upper-triangular

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

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

B Laplacian facts

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

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

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

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

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

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

Recall that, when G is connected, 0 is an eigenvalue of both L and $\mathcal{L}(G)$, with multiplicity $\mathrm{cc}(G)$. Moreover, if G is the union of disjoint graphs $G_1, G_2, \ldots, G_k$, then it has as its spectrum the union of the spectra $\Lambda(G_1), \Lambda(G_2), \ldots, \Lambda(G_k)$. Certain parts of the spectrum of $\mathcal{L}$ can be deduced explicitly for very structured types of G, such as complete graphs, complete bipartite graphs, star graphs, path graphs, and cycle graphs, and n-cubes. For a list of additional properties the graph and normalized Laplacians satisfy, including bounds on eigenvalues, relation to random walks and rapidly-mixing Markov chains, identities tied to isoperimetric properties of graphs, and explicit connections to spectral Riemannian geometry, see [8] and references within.

C Laplacian `matvec` products

Algorithm 1 Two-pass `matvec` operator for combinatorial up-Laplacians in $O(m(p+1)) \approx O(m)$ time

```
Require: $x \in \mathbb{R}^n$, $f : K \to \mathbb{R}_+$, $h : K^p \to [\,n\,]$
Ensure: $y = \langle L_p^{\mathrm{up}}(K, f), x \rangle$
  function UpLaplacianMatvec(x)
    $y \leftarrow \mathbf{0}$                                    ▷ $(\in \mathbb{R}^n)$
    for $\sigma \in K^{p+1}$ do:
      for $\tau,\, \tau' \in \partial[\sigma]^{(2)}$ do:
        $i, j \leftarrow h(\tau),\ h(\tau')$
        $y_i \mathrel{+}= (-1)^{s_{ij}} \cdot x_j \cdot f(\tau) \cdot f(\tau') \cdot f(\sigma)^2$
        $y_j \mathrel{+}= (-1)^{s_{ij}} \cdot x_i \cdot f(\tau) \cdot f(\tau') \cdot f(\sigma)^2$
    for $\sigma \in K^{p+1}$ do:
      for $\tau \in \partial[\sigma]$ do:
        $i \leftarrow h(\tau)$
        $y_i \mathrel{+}= x_i \cdot f(\sigma)^2 \cdot f(\tau)^2$
    return $y$
```