Spectral relaxation of the persistence rank invariant

Matt Piekenbrock
piekenbrock.m@northeastern.edu
Northeastern University

Jose Perea
j.pereabenitez@northeastern.edu
Northeastern University

Abstract

Using a duality result between persistence diagrams and persistence measures, we introduce a framework for constructing families of continuous relaxations of the persistent rank invariant for parametrized families of persistence vector spaces indexed over the real line. Like the rank invariant, these families obey inclusion-exclusion, derive 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, such as smoothness and differentiability over the positive semi-definite cone. Surprisingly, by leveraging a connection between stochastic Lanczos quadrature and implicit trace estimation, our proposed relaxation enables the matrix-free, iterative approximation scheme for all of the persistence invariants—Betti numbers, persistent pairs, and cycle representatives—in linear space and cubic time in the size of the filtration.

Keywords Topological Data Analysis · Persistent Homology · Matrix functions

1 Introduction

Persistent homology [1] (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* $\mu_p^{a,b} \in \mathbb{Z}_+$ [2]:

$$\begin{aligned} \operatorname{dgm}_p(K, f) &= \big\{(a, b) : \mu_p^{a,b} \neq 0\big\} \\ \mu_p^{a,b} &= \min_{\delta>0} \big(\beta_p^{a+\delta,b-\delta} - \beta_p^{a+\delta,b+\delta}\big) - \big(\beta_p^{a-\delta,b-\delta} - \beta_p^{a-\delta,b+\delta}\big) \end{aligned} \tag{1}$$

The surprising and essential quality of persistence is that these pairings exist, are unique, and are stable under additive perturbations [3]. Whether for shape recognition [4], dimensionality reduction [5], or time series analysis [6], 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 [7], persistence landscapes [8], and template functions [9]. Tackling the issue of injectivity, Turner et al. [10] 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 [11].

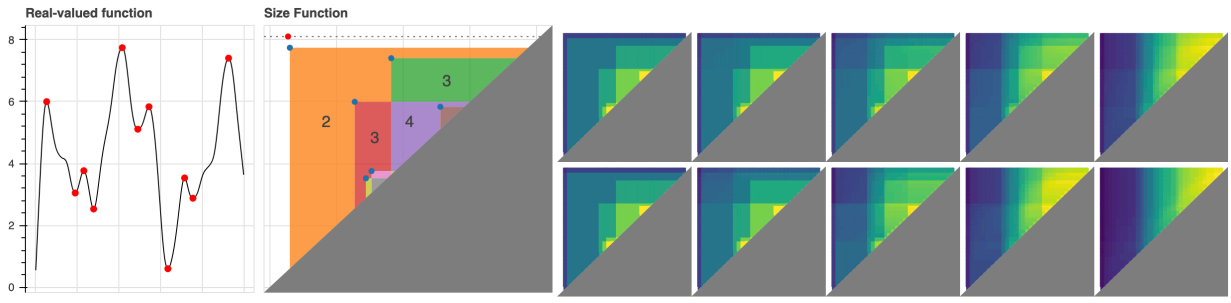

Figure 1: (left) A function $f : \mathbb{R} \to \mathbb{R}$ and its corresponding persistence diagram obtained by filtering f by its sublevel sets. (right) Two spectral interpolations of the rank function to certain Laplacian norms.

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 [2], [12], which generalize (1) to rectangles $R = [a, b] \times [c, d] \subset \mathbb{R}^2$:

$$\mu_p^R(K, f) = \operatorname{card}\big(\operatorname{dgm}_p(K, f)|_R\big) = \beta_p^{b,c} - \beta_p^{a,c} - \beta_p^{b,d} + \beta_p^{a,d} \tag{2}$$

Notably, our approach not only avoids explicitly constructing diagrams, but is also *matrix-free*, circumventing the reduction algorithm from [13] 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, and differentiable on the positive semi-definite cone (Proposition 2). By a reduction to spectral methods for Laplacian operators (Section 2.3), we also show these approximations are computable in $O(m)$ memory and $O(mn)$ time, where n, m are the number of $p, p+1$ simplices in K, respectively (Proposition 3). Moreover, both relaxations admit iterative $(1 \pm \epsilon)$-approximation schemes (Section 3.2), recovering both invariants ϵ is made small enough.

2 Notation & Background

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

$$\partial_p[i, j] \triangleq \begin{cases} (-1)^{s_{ij}} & \sigma_i \in \partial\big[\sigma_j\big] \\ 0 & \text{otherwise} \end{cases}, \qquad \partial_p[\sigma] \triangleq \sum_{i=1}^{p+1} (-1)^{i-1}\big[v_1, ..., v_{i-1}, v_{i+1}, ...v_{p+1}\big] \tag{3}$$

where $s_{ij} = \operatorname{sgn}\big([\sigma_i], \partial\big[\sigma_j\big]\big)$ records the orientation. Extending (3) 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 $K_a \subseteq ... \subseteq K_b$ 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 [14]: 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:

$$H_p^{a,b} = \begin{cases} H_p(K_a) & a = b \\ \operatorname{Im} h_p^{a,b} & a < b \end{cases}, \qquad \beta_p^{a,b} = \begin{cases} \beta_p(K_a) & a = b \\ \dim\big(H_p^{a,b}\big) & a < b \end{cases} \tag{4}$$

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:

$$f_\alpha(\tau) \leq f_\alpha(\sigma) \;\forall\, \tau \subseteq \sigma \in K \;\text{ and }\; f_\alpha(\sigma) \;\text{ is continuous in }\; \alpha \in \mathcal{A} \;\text{ for every }\; \sigma \in K \tag{5}$$

Intuitively, when $\mathcal{A} = \mathbb{R}$, one can think of α as a *time* parameter 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 $f : K \to \mathbb{R}$, note (K, f_α) generalizes the "static" notion of a filtration (K, f) in the sense one may declare $f_\alpha(\sigma) = f(\sigma)$ for all $\alpha \in \mathcal{A}$ and all $\sigma \in K$.
- (Dynamic Metric Spaces) For a set X, let $\gamma_X = (X, d_X(\cdot))$ denote a dynamic metric space [15], 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.*
- (Interpolating filtrations) 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. More generally, any homotopy between filtrations may be used.
- (Fibered barcode) In the 2-parameter persistence setting, a common invariant is the fibered barcode, which induced by projecting a bifiltration to spaces of affine lines with strictly positive slope.

2.1 Technical background

The following results summarize some technical observations motivating this effort, which will be used in several proofs. Though these observations serve as 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 [14], which shows 1-parameter persistence modules decompose in an *essentially unique* way. Computationally, the corresponding Pairing

Uniqueness Lemma [16] 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:

$$R[i,j] \neq 0 \Leftrightarrow \operatorname{rank}(\partial^{i,j}) - \operatorname{rank}(\partial^{i+1,j}) + \operatorname{rank}(\partial^{i+1,j\text{-}1}) - \operatorname{rank}(\partial^{i,j\text{-}1}) \neq 0 \tag{6}$$

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 (6) 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:

$$\operatorname{rank}(R^{i,j}) = \operatorname{rank}(\partial^{i,j}) \tag{7}$$

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 [17], though the latter was also noted in passing by Edelsbrunner [1]. 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 [18]:

Corollary 1 (Bauer et al. [18]): 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 (7), both β_p^* and μ_p^* may be written as a sum of ranks of submatrices of ∂_p and ∂_{p+1}:

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

$$\beta_p^{a_i,a_j}(K, f) = \operatorname{rank}\big(C_p(K_i)\big) - \operatorname{rank}\big(\partial_p^{1,i}\big) - \operatorname{rank}\Big(\partial_{p+1}^{1,j}\Big) - \operatorname{rank}\Big(\partial_{p+1}^{i+1,j}\Big) \tag{8}$$

$$\mu_p^R(K, f) = \operatorname{rank}\Big(\partial_{p+1}^{j+1,k}\Big) - \operatorname{rank}\Big(\partial_{p+1}^{i+1,k}\Big) - \operatorname{rank}\Big(\partial_{p+1}^{j+1,l}\Big) + \operatorname{rank}\Big(\partial_{p+1}^{i+1,l}\Big) \tag{9}$$

Though (9) was pointed out by Cohen-Steiner et al. in [16] and exploited computationally by Chen & Kerber in [19], to the authors knowledge the only explicit derivation and proof of (8) is given by Dey & Wang [17]. For completeness, we give our own detailed proof of Corollary 2 in Section 6.1. 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 [12], the former suggests variational perspectives of the rank function might yield interesting spectral relaxations of (8) and (9) useful for e.g. optimization purposes. Both the latter property with Corollary 1 suggests a potentially new means of computing persistence information *without* matrix reduction. Moreover, combining these observations suggests advances made in other areas of applied mathematics may be readily exploited, such as the rich theory of matrix functions [20], the tools developed as part of "The Laplacian Paradigm" [21], or the recent connections between rank and trace estimation [22]. The rest of the paper is dedicated to exploring these connections and their implications.

2.2 Spectral relaxation and its implications

Before 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:

$$\begin{aligned}\beta_p^{a,b} &= \dim\big(Z_p(K_a) \,/\, B_p(K_b)\big) \\ &= \dim\big(Z_p(K_a) \,/\, \big(Z_p(K_a) \cap B_p(K_b)\big)\big) \\ &= \dim\big(Z_p(K_a)\big) - \dim\big(Z_p(K_a) \cap B_p(K_b)\big)\end{aligned} \tag{10}$$

Computationally, observe that (10) 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 [20] or persistence-based [14], [23] 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 (8) and (9) throughout the rest of the paper.

2.2.1 Parameterized boundary operators

In typical dynamic persistence settings (e.g. [16]), 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:

$$\operatorname{rank}(X) = \operatorname{rank}(P^T X P) \tag{11}$$

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 (8) and (9) 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:

$$\partial_p(\alpha)[k,l] = \begin{cases} s_{kl} \cdot f_\alpha(\sigma_k) \cdot f_\alpha(\sigma_l) & \text{if } \ \sigma_k \in \partial_p(\sigma_l) \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

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 (12) may be decoupled into a product of diagonal matrices $D_*(f_\alpha)$:

$$\partial_p(\alpha) \triangleq D_p(f_\alpha) \cdot \partial_p(K_\preceq) \cdot D_{p+1}(f_\alpha) \tag{13}$$

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 $K_\preceq^p$ and $K_\preceq^{p+1}$, respectively. Clearly, $\operatorname{rank}\big(\partial_p(\alpha)\big) = \operatorname{rank}\big(\partial_p(K_\preceq)\big)$ 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 $\bar{S}_a(x) = \mathbb{1}_{x>a}$ and $S_b(x) = \mathbb{1}_{x\le b}(x)$ to D_p and D_{p+1}, respectively:

$$\hat{\partial}_p^{a,b}(\alpha) \triangleq D_p\big(\bar{S}_a \circ f_\alpha\big) \cdot \partial_p(K_\preceq) \cdot D_{p+1}(S_b \circ f_\alpha) \tag{14}$$

Though these step functions are discontinuous at their chosen thresholds a and b, we may retain the element-wise continuity of (13) 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).

[1] An alternative means to perform matrix reduction is through the PLU factorization (see [17], section 5.1.1), which takes $O(n^\omega)$ time where $\omega \in [2, 2.373)$ is the matrix-multiplication constant. However, this approach exploits neither the structure of the boundary matrix nor properties of persistence to accelerate the computation and is not used in practice.

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 (2) the $\mathcal{A}$-parameterized invariants $\beta_p^{a,b} : \mathcal{A} \times K \to \mathbb{N}$ and $\mu_p^R : \mathcal{A} \times K \to \mathbb{N}$ defined by:

$$\beta_p^{a,b}(\alpha) = \operatorname{rank}\big(D_p(S_a \circ f_\alpha)\big) - \operatorname{rank}\big(\hat{\partial}_p^a(\alpha)\big) - \operatorname{rank}\big(\hat{\partial}_{p+1}^b(\alpha)\big) + \operatorname{rank}\big(\hat{\partial}_{p+1}^{a+\delta,b}\big)(\alpha) \tag{15}$$

$$\mu_p^R(\alpha) = \operatorname{rank}\big(\hat{\partial}_{p+1}^{b+\delta,c}\big) - \operatorname{rank}\big(\hat{\partial}_{p+1}^{a+\delta,c}\big) - \operatorname{rank}\big(\hat{\partial}_{p+1}^{b+\delta,d}\big) + \operatorname{rank}\big(\hat{\partial}_{p+1}^{a+\delta,d}\big) \tag{16}$$

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

Note that in (14), 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 (8) and (9) with (15) and (16), respectively, requires no such explicit filtering.

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

2.3 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:

$$\operatorname{rank}(X) = \operatorname{rank}(XX^T) = \operatorname{rank}(X^T X), \quad \text{for all} \quad X \in \mathbb{F}^{n \times m} \tag{17}$$

In the context of boundary operators, note that $\partial_1 \partial_1^T$ is the well known *graph Laplacian*—more generally, (17) suggests we can readily express $\beta_p^*(\alpha)$ and $\mu_p^*(\alpha)$ using the ranks of combinatorial p-Laplacians.[2]

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

$$\Delta_p(K) = \underbrace{\partial_{p+1} \circ \partial_{p+1}^T}_{L_p^{\text{up}}} + \underbrace{\partial_p^T \circ \partial_p}_{L_p^{\text{dn}}} \tag{18}$$

All three operators Δ_p, L_p^{up}, and L_p^{dn} are symmetric, positive semi-definite, and compact [23]—moreover, the *non-zero* multisets $\Lambda\big(L_p^{\text{up}}\big)$ and $\Lambda\big(L_{p+1}^{\text{dn}}\big)$ 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:

$$\mathcal{L}_{p,\preceq}^{a,b}(\alpha) \triangleq D_p\big(\bar{S}_a \circ f_\alpha\big) \cdot \partial_{p+1}(K_\preceq) \cdot D_{p+1}(S_b \circ f_\alpha) \cdot \partial_{p+1}^T(K_\preceq) \cdot D_p\big(\bar{S}_a \circ f_\alpha\big) \tag{19}$$

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

where $D_p(f)$ denotes a diagonal matrix whose entries represent the application of f to the p-simplices of K. As in (14), 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 $i = \sum_{\sigma \in K}\big(\bar{S}_a \circ f_\alpha\big)(\sigma)$ and $j = \sum_{\sigma \in K}(S_b \circ f_\alpha)(\sigma)$ such that:

$$\operatorname{rank}\Big(R_{p+1}^{i,j}\Big) = \operatorname{rank}\Big(\partial_{p+1}^{i,j}\Big) = \operatorname{rank}\Big(\hat{\partial}_{p+1}^{a,b}\Big) = \operatorname{rank}\bigg(\Big(\hat{\partial}_{p+1}^{a,b}\Big)\Big(\hat{\partial}_{p+1}^{a,b}\Big)^T\bigg) = \operatorname{rank}\Big(\mathcal{L}_{p,\preceq}^{a,b}\Big) \tag{20}$$

where the second last equality uses the identity $\operatorname{rank}(X) = \operatorname{rank}(X^T X)$, which is true when $X \in \mathbb{F}^{n \times m}$ has coefficients in a zero-characteristic field $\mathbb{F}$. 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.

It is worth noting that composing step functions $\bar{S}_a, S_b : \mathbb{R} \to [0, 1]$ with f_α is equivalent to endowing a *weight function* $w : K \to (0, +\infty)$ on a subset $K_{a,b} \subseteq K$, in the sense described by [23]. In particular, if w_p denotes the restriction of w to K^p, then w_p defines an inner product $\langle \cdot, \cdot \rangle_{w_p}$ on space of p-chains $C_p(K, \mathbb{R})$ given by:

$$\langle [\sigma], [\sigma'] \rangle_{w_p} \triangleq \delta_{\sigma\sigma'} \cdot \big(w_p(\sigma)\big)^{-1}, \quad \forall\, \sigma, \sigma' \in K^p \tag{21}$$

where $\delta_{\sigma\sigma'} = 1(\sigma = \sigma')$ is the indicator function on σ. Moreover, this inner product $\langle [\sigma], [\sigma'] \rangle_{w_p}$ on $C_p(K, \mathbb{R})$ induces an inner product:

$$\langle\!\langle f, g \rangle\!\rangle_w = \sum_{\sigma \in K^p} f([\sigma]) g([\sigma]) w(\sigma), \quad \text{for all} \quad f, g \in C^p(K) \tag{22}$$

In these sense above, (19) is simply choosing a particular inner product on the space of p-cochains.

Remark: 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. (10) and (19) could be alternatively defined using the inclusion maps $L \hookrightarrow K$ between *simplicial pairs* (L, K), as in [23].

2.4 Spectral rank relaxation

Under mild assumptions on f_α, the entries of the boundary operators from (14) are continuous functions of α when S is substituted appropriately with smoothstep functions. In contrast, the quantities from Proposition 1 are by definition discontinuous functions, as they are integer-valued due to the rank function. To circumvent this issue, we consider the spectral characterization of the rank function:

$$\operatorname{rank}(X) = \sum_{i=1}^{n} \operatorname{sgn}_+(\sigma_i(X)), \qquad \operatorname{sgn}_+(x) = \begin{cases} 1 & \text{if} \;\; x > 0 \\ 0 & \text{otherwise} \end{cases} \tag{23}$$

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 [20], it is clear the discontinuity in (23) manifests from the one-sided sign function—thus, a natural approach to relaxing (23) is to first relax the sgn_+ function.

Our approach follows the seminal work of Mangasarian et al. [25]. 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:

$$\phi(x, \tau) = \int_{-\infty}^{x} \hat{\delta}(z, \tau)\, \mathrm{d}z, \quad \hat{\delta}(z, \tau) = \frac{1}{\nu(\tau)} \cdot p\bigg(\frac{z}{\nu(\tau)}\bigg), \quad \forall z \geq 0, \tau > 0 \tag{24}$$

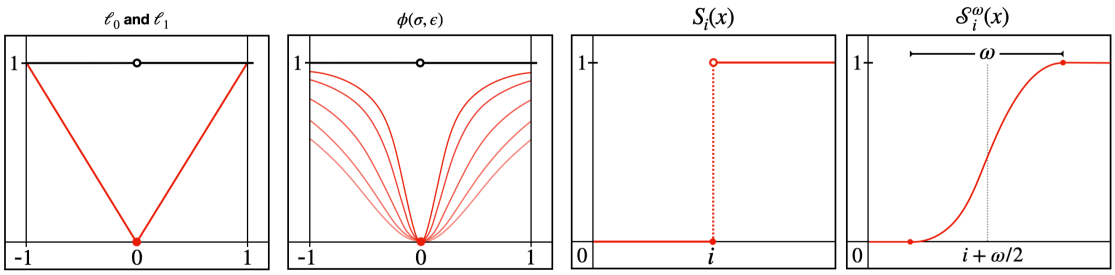

Figure 2: From left to right—the ℓ_1 (red) norm and ℓ_0 (black) pseudo-norm are shown on the interval $[-1, 1]$; the relaxation $\phi(\cdot, \tau)$ from (28) at various values of $\tau > 0$ (red) and at $\tau = 0$ (black); the step function $S_i(x)$ from (14); the smoothstep relaxation $\mathcal{S}_i^{\omega}$ discussed in Section 2.2.1.

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 (24) 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 (24) extend naturally to the rank function when substituted appropriately via (23). 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 [26]:

$$\Phi_\tau(X) \triangleq \sum_{i=1}^{n} \phi(\sigma_i, \tau) u_i v_i^T \tag{25}$$

Note that when X is positive semi-definite, (25) may be interpreted as a particular choice of *matrix function* $f(X) \triangleq U f(\Lambda) U^T$ from the matrix function calculus perspective [20]. Unlike general matrix functions, however, the restrictions on ϕ (24) grants the operators Φ_τ a variety of attractive properties related to approximation, monotonicity, and differentiability.

Proposition 2 (Bi et al. [26]): The operator $\Phi_\tau : \mathbb{R}^{n \times m} \to \mathbb{R}^{n \times m}$ defined by (25) 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 $r = \text{rank}(X)$ and positive singular values $\Lambda(X) = \{\sigma_1, \sigma_2, ..., \sigma_r\}$:*

$$0 \leq r - \|\Phi_\tau(X)\|_* \leq r \cdot (1 - \phi(\sigma_r, \tau)) \tag{26}$$

   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 (14) are smoothed ($\omega > 0$).

Remark: 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 [26]. $\|\Phi_\tau(X)\|_*$ is differentiable on the positive semi-definite cone $\mathbb{S}_+^n$.

[3] *Here, "semismooth" refers to the existence certain directional derivatives in the limit as $\tau \to 0^+$, see [20], [20].*

[4] It may be shown to be twice continuously differentiable at X if ϕ is twice-differentiable at each $\sigma_i(X)$, see [27]

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 $\tau > 0$:

$$x_\tau^* = \arg\min_{x\in\mathbb{R}^n} \|Ax - b\|^2 + \tau\|x\|^2 = \left(A^TA + \tau I\right)^{-1} A^Tb \tag{27}$$

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 (24):

$$\phi(x,\tau) = \int_0^z \hat{\delta}(z,\tau)dz = \frac{2}{\tau}\int_0^z z \cdot \left(\left(z/\sqrt{\tau}\right)^2 + 1\right)^{-2} dz = \frac{x^2}{x^2+\tau} \tag{28}$$

By substituting $\mathrm{sgn}_+ \mapsto \phi$ and composing with the singular value function (25), the corresponding spectral rank approximation reduces[5] to the following *trace* formulation:

$$\|\Phi_\tau(A)\|_* = \sum_{i=1}^n \frac{\sigma_i(A)^2}{\sigma_i(A)^2+\tau} = \mathrm{tr}\Big[\left(A^TA + \tau\, I\right)^{-1} A^TA\Big] \tag{29}$$

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 (25) 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*:

$$v'(t) = -L_G v(0) \quad \Leftrightarrow \quad L_G \cdot u(x,t) = -\partial u(x,t)/\partial t \tag{30}$$

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:

$$v(t) = H_t\, v(0), \text{ where } \; H_t = \sum_{i=1}^n e^{-t\lambda_i}\, u_i\, u_i^T \tag{31}$$

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 (31), it is also a matrix function. Now, consider (24) with $\nu(\tau) = \tau$ and $p(\lambda) = \exp(-\lambda_+)$ where $x_+ = \max(x, 0)$:

$$\phi(\lambda,\tau) = \int_0^z \hat{\delta}(z,\tau)dz = \frac{1}{\tau}\int_0^z \exp(-z/\tau)dz = 1 - \exp(-\lambda/\tau), \quad \text{for all } \; \lambda \geq 0 \tag{32}$$

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 (25) again yields an equivalent trace expression:

$$\|\Phi_\tau(L_G)\|_* = \sum_{i=1}^n 1 - \exp(-\lambda_i/\tau) = n - \mathrm{tr}\Big[H_{1/\tau}\Big] \tag{33}$$

[5] See Theorem 2 of [28] for a proof of the second equality.

[6] This can be seen directly by (27) as well, wherein increasing τ lowers the condition number of $A^TA + \tau I$ monotonically, signaling a tradeoff in stability at the expense of accuracy.

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 (32) as a *geometrically informative RI approximation*.

3 Computational Implications

3.1 Exact computation

As evidenced by Section 2.2, 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* [29], which estimates the eigenvalues of any symmetric linear operator A via projection onto successive Krylov subspaces. Given a symmetric $A \in \mathbb{R}^{n\times n}$ with eigenvalues $\lambda_1 \geq \lambda_2 > ... \geq \lambda_r > 0$ and a vector $v \neq 0$, the Lanczos method generates the triple (K, Q, T):

$$\begin{aligned} K &= [\, A^0 v \mid A^1 v \mid A^2 v \mid ... \mid A^{r-1} v \,] \\ Q &= [\, q_1, q_2, ..., q_r \,] \leftarrow \mathrm{qr}(K) \\ T &= Q^T A Q \end{aligned} \tag{34}$$

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 its only prerequisite for computation being a matrix-vector product operator $v \mapsto Av$—A need not necessarily be stored in memory explicitly. For this reason, the Lanczos method is often used as a low-memory option for computing eigenvalues. Indeed, due to its *three-term recurrence* [30], the Lanczos method requires just three $O(n)$-sized vectors and a constant number of $O(n)$ vector operations to obtain T—neither K nor Q need be formed explicitly.

Lemma 2 ([31]): 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 $\Lambda(A) = \{\lambda_1, \lambda_2, ..., \lambda_r\}$ in $O(\max\{\eta, n\} \cdot r)$ time and $O(\max\{\nu, n\})$ space, when computation is done in exact arithmetic.

Surprisingly, Lemma 2 suggests the time and space complexities of computing quantities that depend on $\{\, \lambda_1, \lambda_2, ..., \lambda_r \,\}$ are lower for certain classes of matrices, provided ν and η are sufficiently small. In general, the size of these variables depends on the structure and sparsity of the operators[7] involved. In some cases, both ν and η are $O(n)$; for example, the application $x \mapsto Lx$ for the graph Laplacian $L = \partial_1 \partial_1^T$ is linear in the number of edges $|E|$ due to its graph structure. Though this 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 $m = \left|K^{p+1}\right|$, if there exists a hash function $h : K^p \to [n]$ constructible in $O(m)$ time supporting $O(1)$ access time and $O(c)$ space, then there exists a two-phase algorithm for computing the 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 Section 6. 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

[7] For $n \times n$ sparse matrices with an average of z non-zeros per row and rank r, for example, Lemma 2 implies the Lanczos method has an expected $O(nzr)$ time and $O(nz)$ space complexities [32].

delegate more practical issues regarding the computation to Algorithm 1. Combining Lemma 2 and Lemma 3 yields our main result for this section.

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 $n_{ad} = \left|K_d^p\right| - \left|K_a^p\right|$ and $m_{ad} = \left|K_d^{p+1}\right| - \left|K_a^{p+1}\right|$, can be computed in exact arithmetic with the *Lanczos method* in the following time and space complexities:

$$\mu_p^R(K) \overset{\text{time}}{=} O(n_{ad} \cdot m_{ad}), \quad \mu_p^R(K) \overset{\text{space}}{=} O(\max(n_{ad}, m_{ad})) \tag{35}$$

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 $n = \left|K^p\right|$ and $m = \left|K^{p+1}\right|$.

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 $N = |K| \sim O\left(\left|K^{p+1}\right|\right)$ in $\Theta(N^3)$ time and $\Theta(N^2)$ space. Interestingly, Chen and Kerber [19] 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 $\left|K^p\right|$ and $\left|K^{p+1}\right|$ are trivially bounded 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: A similar result for computing *non-persistent* Betti numbers of simplicial complexes over finite fields was given by Edelsbrunner and Parsa in [33], wherein the complexity of computing the Betti numbers of a 2-dimensional simplicial complex K with n vertices was shown to be $\Omega(r(n, m))$, where $r(n, m)$ is the complexity of computing the rank of an n-by-n binary matrix with m non-zero entries.

3.2 Randomized (η, ϵ)-approximation

As in [31], 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 affect both the convergence and termination conditions of the Lanczos method, prohibiting its use practically. Though many improvements have been proposed throughout the decades (e.g. selective re-orthogonalization, implicit restarting, see [34], [35] for an overview), it is well-known that the Lanczos method is not efficient at accurately approximating eigenvalues on the interior of the spectrum.

Fortunately, it turns out accurately estimating eigenvalues is not necessary for estimating the rank. Recently, it has been shown that the Lanczos method is stable for matrix function approximation[8] even in the finite precision arithmetic model. The use of Lanczos for matrix function approximation is motivated by the fact that Lanczos expansion up to degree k can *exactly* apply any matrix polynomial p with degree $< k$ [36]:

$$p(A)q_1 = Qp(T)e_1 \Leftrightarrow \quad f(A)x = \|x\| \cdot Qf(T)e_1 \tag{36}$$

In particular, Paige's A27 Lanczos variant ([37]), when executed up to degree k, has an error that is bounded by the uniform error the best degree-p polynomial approximation to any bounded function f with degree $p < k$ (see [36] for conditions). For general matrix functions $f(A)$, this implies that finite-precision Lanczos essentially matches the strongest known exact arithmetic bounds. The particular relevance of (36) to the computation of our proposed spectral ϕ-approximation (25) is the fact that the quantity of interest $\left\|\Phi\left(\mathcal{L}_p\right)\right\|_*$ is expressible as a trace-norm, i.e. it may be computed via:

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

$$\|\Phi_\tau(A)\|_* = \mathrm{tr}\big(\Phi_\tau(\mathcal{L}_p)\big) = \sum_{i=1}^{n} e_i^T \Phi_\tau(\mathcal{L}_p) e_i \tag{37}$$

where $\{e_1, ..., e_n\}$ is an orthonormal basis. For this reason, we propose the use *stochastic Lanczos quadrature* (SLQ) type methods [22] for estimating quantities of the form $\mathrm{tr}(f(A))$. The simplest such estimator is the Girard-Hutchinson (GH) estimator:

$$\mathrm{tr}(f(A)) \approx \frac{n}{n_v} \sum_{j=1}^{n_v} e_1^T f\Big(T_k^{(j)}\Big) e_1 = \frac{n}{n_v} \sum_{j=1}^{n_v} \left(\sum_{i=1}^{k} \tau_i^{(j)} f\Big(\theta_i^{(j)}\Big) \right), \quad \tau_i = \big[e_1^T y_i\big]^2 \tag{38}$$

where $T_k^{(j)}$ represents the result applying Lanczos to the degree-$(k+1)$ Krylov expansion of $\big(A, v^{(j)}\big)$ and (θ_i, y_i) represent the Rayleigh-Ritz pairs associated with the tridiagonal eigendecomposition $T_k = Y \Theta Y^T$. When $v^{(j)} \sim \mathcal{D}$ derives from a sub-Gaussian distribution satisfying $\mathbb{E}\big[v^{(j)} \otimes v^{(j)}\big] = I$, the approximation (38) is known to be an unbiased estimator of $\mathrm{tr}(f(A))$. Under mild assumptions (see [22]), 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 GH estimator Γ satisfies:

$$\Pr(|\mathrm{tr}(f(A)) - \Gamma| \leq \epsilon |\mathrm{tr}(f(A))|) \geq 1 - \eta \tag{39}$$

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 $e_1^T f(T_k) e_1$, where each T_k is generated by applying degree-$(k+1)$ Lanczos to pairs (A, v) generated over random $v \sim \mathcal{D}$. In Section 5.1, we show how to extend this result to the persistence setting.

More recent work by [38] has shown that deflation techniques can reduce the number of matrix-vector evaluations needed down to $O\Big(\varepsilon^{-1} \cdot \sqrt{\log(\eta^{-1})} + \log(\eta^{-1})\Big)$, though we note in Section 5.1 there are additional limitations to be aware of using these methods (e.g. a $O(mk)$ space complexity).

3.3 Apparent pairs optimization

One of the defining aspects of the above trace-based computation is that, unlike the reduction algorithm, execution does not require matrix decomposition. Unfortunately, one downside to this is that we lose access to certain computational shortcuts developed specifically for the reduction setting; it is not immediately clear whether persistence-specific optimizations like *clearing* and *cohomology* [17] have analogous shortcuts in the matrix-free setting. Nonetheless, one such optimization well known for clique filtrations—the identification of *apparent pairs*—does have a direct translation to the trace estimation setting. On some types of problems, this optimization alone enables us to discard 90-99% of the columns of ∂_p prior to any rank computations.

Apparent pairs (APs) are a class of persistence pairs which are already reduced in the filtration boundary matrix $\partial(K)$. To motivate their use in our proposed trace-based computation, consider the following 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 the columns of $\partial_{p+1}(K)$ into submatrices ∂_{p+1}^- and ∂_{p+1}^+:

$$\partial_{p+1}^- = \{\, \partial\big[\sigma_j\big] : \mathrm{col}_R(j) \neq 0, \sigma_j \in K^{p+1} \,\}, \quad \partial_{p+1}^+ = \{\, \partial\big[\sigma_j\big] : \mathrm{col}_R(j) = 0, \sigma_j \in K^{p+1} \,\} \tag{40}$$

and so that:

$$\mathrm{rank}\big(\partial_{p+1}(K)\big) = \mathrm{rank}\big(\partial_{p+1}^-(K)\big) \tag{41}$$

In other words, from a rank-based perspective, we may safely discard $p+1$ simplices whose boundary chains lie in the kernel of R. Of course, if ∂_{p+1}^- if obtained from the full decomposition $R = \partial K$, then the rank of any submatrix of ∂ is fully determined, leaving the utility of the above observation moot.

Fortunately, for simplexwise clique filtrations, we can use APs $(\tau, \sigma) \in \mathrm{dgm}_{p+1}(K)$ to construct an approximation $\hat{\partial}^{-}_{p+1} \supset \partial^{-}_{p+1}$ efficiently. The main benefit of using APs here is that they can be readily identified based on a purely local condition, which is evident from their very definition:

Definition (Apparent Pair): Given a simplexwise filtration $K_{\preceq}$ of a 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.

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. Since any persistent pair $(\tau, \sigma) \in \mathrm{dgm}_p(K)$ corresponds to a pair of columns in R_p and R_{p+1} (respectively), the boundary chain $\partial[\sigma]$ must lie in the kernel of $\partial_{p+1}V_{p+1}$, and therefore can be safely discarded. 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 [39], which restricts its computation to simplexwise filtrations induced by the *reverse colexicographical* vertex order. In this setting, p-simplices $\left(v_{i_p}, ..., v_{i_0}\right)$ satisfying $v_{i_p} > ... > v_{i_0}$ are mapped to integers $r \in [0, C(n, p+1))$—where $C(n,k)$ is the binomial coefficient—via the *combinatorial number system*; when the colexicographical order is used, the bijection is given by the sum $\left(v_{i_p}, ..., v_{i_0}\right) \mapsto \sum_{j=1}^{p+1} C(i_j, j)$, and cofacets are given by the following relation:

$$\left(v_{i_p}, ..., v_{i_{k+1}}, v_j, v_{i_k}, ..., v_{i_0}\right) \mapsto \sum_{l=k+1}^{p} \binom{i_l}{l+2} + \binom{j}{k+1} + \sum_{l=0}^{k} \binom{i_l}{l+1} \tag{42}$$

By enumerating $j = n-1, ..., 0$ for $j \notin [n] \setminus \{i_p, ..., i_0\}$, one recovers all of $n-(p+1)$ cofacets of $\left(v_{i_p}, ..., v_{i_0}\right)$ 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 (42) 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 [39]). 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 [39] 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.

4 Applications & Experiments

4.1 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 [3], 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

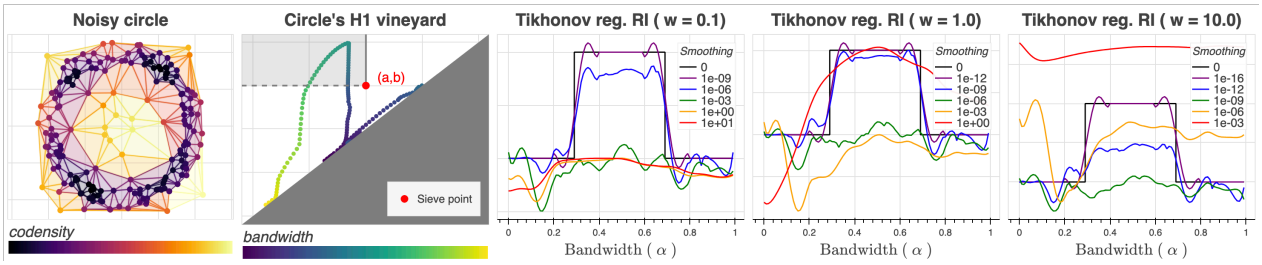

Figure 3: From left to right: Delaunay complex K realized from point set $X \subset \mathbb{R}^2$ sampled with multiple types of noise around S^1 (colored by codensity at optimal $\alpha^* \approx 1/2$); codensity vineyard of (K, f_α) across varying bandwidths α and a fixed point $(a, b) \in \Delta_+$;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

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:

$$\alpha^* = \arg\max_{\alpha \in \mathbb{R}} \beta_p^{a,b}(K, f_\alpha), \quad \text{where} \quad f_\alpha(x) = \frac{1}{n\alpha} \sum_i C(\mathcal{K})) - \mathcal{K}_\alpha(x_i - x) \tag{43}$$

where $C(\mathcal{K}_\alpha)$ is a normalizing constant that depends on the choice of 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 $(a^*, b^*) \in \mathrm{dgm}_1(K, f_{\alpha^*})$. Thus, if we place a corner point (a, b) satisfying $a^* \leq a$ and $b < b^*$ for some $\alpha > 0$, we expect $\beta_p^{a,b}(\alpha) = 1$ to near the optimal bandwidth α^*—matching the first Betti number of S^1—and 0 otherwise.

In Figure 3, we depict the dimension-1 vineyard 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 (43), enabling the use of first-order optimization techniques. By using the Tikhonov regularization from (29), 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. 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. 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.

4.2 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 (e.g. curvature, co-planarity) are now well known (see [40] for an overview), *topology-preserving* simplification techniques are relatively sparse, especially for higher embedding dimensions. Moreover, such procedures typically restrict to operations that preserve *local* notions of topology, such as the genus of a feature's immediate neighborhood or the property of being a manifold. These operations 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. In contrast to [41], we forgo the use of persistence-preserving operations and

instead opt for a simpler strategy: we perform an exponential search on a given sequence of simplifications, settling on the largest simplification found.

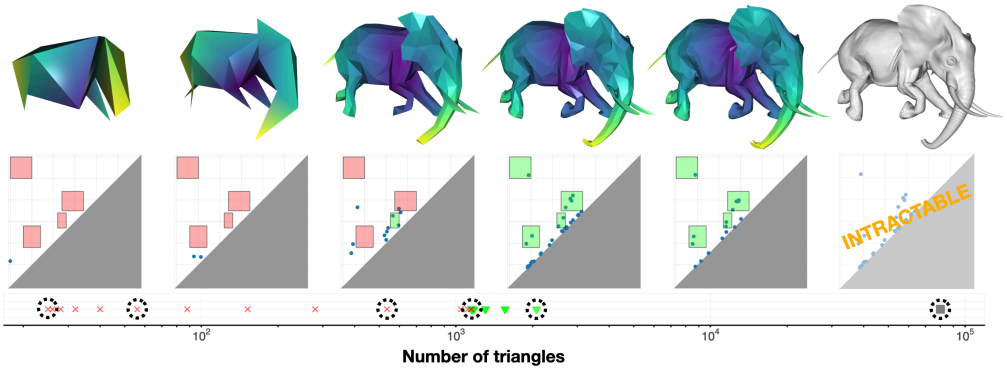

Figure 4: (Top) Meshes filtered and colored by eccentricity at varying levels of simplification; (middle) their diagrams and topological constraints; (bottom) simplification thresholds tested by an exponential search, on a logarithmic scale. The color/shape of the markers indicate whether the corresponding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

We show an exemplary application of this idea in Figure 4 in sparsifying a mesh of an elephant. To filter the mesh in a geometrically meaningful way, we use the (geodesic) *eccentricity* function, which assigns points $x \in X$ in a metric space (X, d_X) a non-negative value representing the distance that point is from the center of the mesh:

$$E_p(x) = \left(\frac{\sum_{x' \in X} d_X(x, x')^p}{N} \right)^{\frac{1}{p}} \tag{44}$$

We may extend $E_p(x)$ to simplices $\sigma \in K$ by taking the max $f(\sigma) = \max_{v \in \sigma} d_X(x, v)$. Note this does not require identifying a center point in the mesh. Intuitively, the highly persistent H_1 classes in mesh carrying the most detail corresponds to "tubular" features that persist from the center; the four legs, the trunk, the two ears, and two tusks. In this example, four rectangular-constraints are given which ensure the simplified elephants retain these features with certain minimal persistence $\delta > 0$.

Note that neither persistence diagrams nor deformation-compatible simplicial maps were needed to perform the sparsification, just a priori knowledge of which areas of Δ_+ to check for the existence of persistent topological features. We defer a full comparison of the sparsification application as future work.

4.3 Topological time series analysis

In many time series applications, detecting periodic behavior can prove a difficult yet useful thing to estimate. For example, in medical images contexts, it is necessary to preprocess the data to remove noise and outliers that otherwise obscure the presence of recurrent behavior.

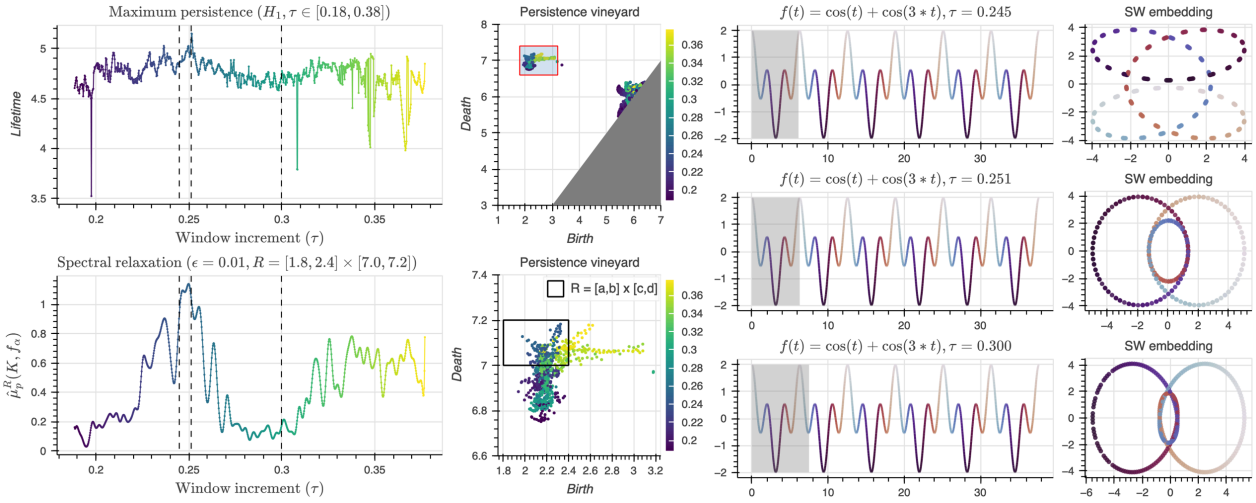

Figure 5: SW1Pers

5 Concluding Remarks and Limitations

In summary, we have introduce various spectral relaxations of the persistent rank function, which has a wealth of interesting properties, including differentiability on the positive semi-definite cone, a matrix-free representation, and natural connections to variational approaches common in machine learning applications, such as the Tikhonov regularization and the Heat kernel. By focusing on coefficients in $\mathbb{R}$, we were able to exploit the connection between the inner product spaces on cochain spaces and the theory of persistence measures, providing an avenue to introduce diffentiability to an otherwise discontinuous function. Moreover, by focusing on the spectral characterization of the rank function, we were able to study persistence in the setting of *iterative methods*, a pursuit which lead us to the Lanczos method of Krylov expansion. Surprisingly, these computational techniques turned out to pair well with the output-sensitive algorithm from [19], which we can use to compute either only Γ-persistence pairs or the full diagram. As cycle representatives can be obtained once a pairing has been constructed, the iterative approach may be used to compute essentially any persistence invariant.

5.1 Limitations

There are few limitations to the proposed work that may prevent its practical use, depending on the problem and invariant of interest. We mention the limitations to the approaches proposed that we are aware of below.

Function parameterization: On the spectral side, in particular, tuning the degree of regularization of the chosen matrix function (25) can be very application-dependent. We demonstrate this in Figure 3 —for some values of $\tau > 0$, the relaxation can be either have high variance due to instability or it can exhibit high bias due over smoothing. In the optimization setting, we addressed these issues via spectral regularization and iterative shrinkage, however other methods could also be used. The reason this phenomenon occurs in general is due to the combinatorial nature of the rank invariant itself: although the rank invariant is often called a "stable function" even in the multi-parameter setting [2], this stability exists **only** in the pairings themselves. Most notably, the *Betti sequence*—which is the sequence containing the Betti numbers of the homology groups at all scales—is **unstable** with respect to the 1-Wasserstein distance (see Theorem 2.5 in [42]).

Randomized estimation: In practice, the efficiency of the iterative implicit trace estimator proposed in Section 3.2 depends on a variety of factors. For some of the persistence invariants—like the pairing itself—

a single wrong rank value in the divide-and-conquer algorithm from [19] can lead to unpredictable output. Thus, if the invariant to be computed needs to be correct with high confidence and the randomized estimator has success probability $p > 0$, then one may need to be re-run k times to achieve a higher success probability $\big(1 - (1-p)^k\big)$. Fortunately, for any constant $\delta \in (0, 1)$, Lemma 10 of [19] shows the number of such rank computations needed to compute Γ-persistent pairs is not larger than:

$$\rho = \rho\bigg(n, \Big\lceil \frac{1}{\delta} \Big\rceil\bigg) = 4n\bigg(2\Big\lceil \frac{1}{\delta} \Big\rceil + \log n + 1\bigg) \tag{45}$$

Indeed, the analysis from section 6 of [19] shows that is the rank estimator has success probability p and $r = 1 - p$, then for the entire algorithm to achieve an arbitrary success probablity $q > 0$, every rank computation needs to be repeated at most k times, where:

$$\log(1/r)^{-1} \cdot \log(\omega) \leq k, \quad \omega = \Big(1 - q^{\frac{1}{\rho}}\Big)^{-1} \tag{46}$$

Since $\log(\omega) = \Theta(\rho)$, we have that every rank computation requires $O(\log \rho) = O\big(\log n + \log \frac{1}{\delta}\big)$ repeated trials. In other words, the randomized aspect of the algorithm does not signicantly affect its scalability.

Estimator efficiency: The primary limitation of approximating the rank function via the GH Monte-carlo estimator via (38) is its convergence, which is dominated by $O(\epsilon^{-2})$. If ϵ is small, the number of iterations the estimator needs can be enormous. Thus its practical utility lies in applications where ϵ need not be too small, i.e. only a relatively coarse approximation of $\operatorname{tr}(f(A))$ is needed. Unfortunately, estimating the rank exactly requires shrinking ϵ down to a factor of $O(1/n)$. This combination of slow convergence and high precision is the primary limitation of the (38) in practical persistence settings, and to us to the only barrier preventing widespread adoption.

Fortunately, implicit trace estimation is active area of research. Recent results by Meyers et al. [38], for example, show how the estimator (38) can be modified to obtain a $(1 \pm \epsilon)$ approximation using only $O\big(\varepsilon^{-1} \cdot \sqrt{\log(\eta^{-1})} + \log(\eta^{-1})\big)$ samples using deflation techniques, though this does require additional re-orthogonalization and storage costs. Even more recently, optimal trace estimators based on the *exchangeability principle* and the *Nyström approximation* have been shown to not only match the $O(1/n^2)$ variance reduction rate Hutch++, but to also empirically achieve much faster convergence [43].

Interestingly, using a reduction to the Gap-Hamming problem, it was shown that achieving a $(1 \pm \epsilon)$ trace approximation with probability greater than $3/4$ requires at least $\Omega(\epsilon^{-1})$ matrix-vector queries $v \mapsto Av$ if the input vectors $v \in \mathbb{R}^n$ are chosen *non-adaptively* [38] (i.e. independent of the structure of A). This lower bound suggests its possible to achieve a $O(1 \pm \epsilon)$ approximation of the rank-based invariants from Proposition 3 with the same space complexity and without exact arithmetic, but no algorithm is known to the author that achieves this non-adaptive bound.

References

[1] H. Edelsbrunner, D. Letscher, and A. Zomorodian, "Topological persistence and simplification," in *Proceedings 41st annual symposium on foundations of computer science*, 2000, pp. 454–463.

[2] A. Cerri, B. D. Fabio, M. Ferri, P. Frosini, and C. Landi, "Betti numbers in multidimensional persistent homology are stable functions," *Mathematical Methods in the Applied Sciences*, vol. 36, no. 12, pp. 1543–1557, 2013.

[3] D. Cohen-Steiner, H. Edelsbrunner, and J. Harer, "Stability of persistence diagrams," in *Proceedings of the twenty-first annual symposium on Computational geometry*, 2005, pp. 263–271.

[4] F. Chazal, D. Cohen-Steiner, L. J. Guibas, F. Mémoli, and S. Y. Oudot, "Gromov-Hausdorff stable signatures for shapes using persistence," in *Computer Graphics Forum*, 2009, pp. 1393–1403.

[5] L. Scoccola and J. A. Perea, "FibeRed: Fiberwise Dimensionality Reduction of Topologically Complex Data with Vector Bundles," in *39th International Symposium on Computational Geometry (SoCG 2023)*, 2023.

[6] J. A. Perea, "Persistent homology of toroidal sliding window embeddings," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 6435–6439.

[7] H. Adams *et al.*, "Persistence images: A stable vector representation of persistent homology," *Journal of Machine Learning Research*, vol. 18, 2017.

[8] P. Bubenik and others, "Statistical topological data analysis using persistence landscapes.," *J. Mach. Learn. Res.*, vol. 16, no. 1, pp. 77–102, 2015.

[9] J. A. Perea, E. Munch, and F. A. Khasawneh, "Approximating continuous functions on persistence diagrams using template functions," *Foundations of Computational Mathematics*, pp. 1–58, 2022.

[10] K. Turner, S. Mukherjee, and D. M. Boyer, "Persistent homology transform for modeling shapes and surfaces," *Information and Inference: A Journal of the IMA*, vol. 3, no. 4, pp. 310–344, 2014.

[11] M. Piekenbrock and J. A. Perea, "Move Schedules: Fast persistence computations in coarse dynamic settings," *arXiv preprint arXiv:2104.12285*, 2021.

[12] F. Chazal, V. De Silva, M. Glisse, and S. Oudot, *The structure and stability of persistence modules*, vol. 10. Springer, 2016.

[13] H. Edelsbrunner and J. L. Harer, *Computational topology: an introduction*. American Mathematical Society, 2022.

[14] A. Zomorodian and G. Carlsson, "Computing persistent homology," in *Proceedings of the twentieth annual symposium on Computational geometry*, 2004, pp. 347–356.

[15] W. Kim and F. Mémoli, "Spatiotemporal persistent homology for dynamic metric spaces," *Discrete & Computational Geometry*, vol. 66, pp. 831–875, 2021.

[16] D. Cohen-Steiner, H. Edelsbrunner, and D. Morozov, "Vines and vineyards by updating persistence in linear time," in *Proceedings of the twenty-second annual symposium on Computational geometry*, 2006, pp. 119–126.

[17] T. K. Dey and Y. Wang, *Computational topology for data analysis*. Cambridge University Press, 2022.

[18] U. Bauer, T. B. Masood, B. Giunti, G. Houry, M. Kerber, and A. Rathod, "Keeping it sparse: Computing Persistent Homology revised," *arXiv preprint arXiv:2211.09075*, 2022.

[19] C. Chen and M. Kerber, "An output-sensitive algorithm for persistent homology," in *Proceedings of the twenty-seventh annual symposium on Computational geometry*, 2011, pp. 207–216.

[20] R. Bhatia, *Matrix analysis*, vol. 169. Springer Science & Business Media, 2013.

[21] S.-H. 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*, 2010, pp. 2–14.

[22] S. Ubaru and Y. Saad, "Fast methods for estimating the numerical rank of large matrices," in *International Conference on Machine Learning*, 2016, pp. 468–477.

[23] F. Mémoli, Z. Wan, and Y. Wang, "Persistent Laplacians: Properties, algorithms and implications," *SIAM Journal on Mathematics of Data Science*, vol. 4, no. 2, pp. 858–884, 2022.

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

[25] O. Mangasarian and C. Chen, "A class of smoothing functions for nonlinear and mixed complementarity problems," 1994.

[26] S. Bi, L. Han, and S. Pan, "Approximation of rank function and its application to the nearest low-rank correlation matrix," *Journal of Global Optimization*, vol. 57, no. 4, pp. 1113–1137, 2013.

[27] C. Ding, D. Sun, J. Sun, and K.-C. Toh, "Spectral operators of matrices," *Mathematical Programming*, vol. 168, no. 1, pp. 509–531, 2018.

[28] Y.-B. Zhao, "An approximation theory of matrix rank minimization and its application to quadratic equations," *Linear Algebra and its Applications*, vol. 437, no. 1, pp. 77–93, 2012.

[29] C. Lanczos, "An iteration method for the solution of the eigenvalue problem of linear differential and integral operators," 1950.

[30] H. D. Simon, "Analysis of the symmetric Lanczos algorithm with reorthogonalization methods," *Linear algebra and its applications*, vol. 61, pp. 101–131, 1984.

[31] B. N. Parlett, "Do we fully understand the symmetric Lanczos algorithm yet," *Brown et al*, vol. 3, pp. 93–107, 1994.

[32] G. H. Golub and C. F. Van Loan, *Matrix computations*. JHU press, 2013.

[33] H. Edelsbrunner and S. Parsa, "On the computational complexity of Betti numbers: reductions from matrix rank," in *Proceedings of the twenty-fifth annual ACM-SIAM symposium on discrete algorithms*, 2014, pp. 152–160.

[34] D. Sorensen, "Implicitly restarted Arnoldi/Lanczos methods and large scale SVD applications," *SVD and Signal Processing III*, pp. 21–31, 1995.

[35] R. B. Lehoucq, D. C. Sorensen, and C. Yang, *ARPACK users' guide: solution of large-scale eigenvalue problems with implicitly restarted Arnoldi methods*. SIAM, 1998.

[36] C. Musco, C. Musco, and A. Sidford, "Stability of the Lanczos method for matrix function approximation," in *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, 2018, pp. 1605–1624.

[37] C. C. Paige, "Computational variants of the Lanczos method for the eigenproblem," *IMA Journal of Applied Mathematics*, vol. 10, no. 3, pp. 373–381, 1972.

[38] R. A. Meyer, C. Musco, C. Musco, and D. P. Woodruff, "Hutch++: Optimal stochastic trace estimation," in *Symposium on Simplicity in Algorithms (SOSA)*, 2021, pp. 142–155.

[39] U. Bauer, "Ripser: efficient computation of Vietoris–Rips persistence barcodes," *Journal of Applied and Computational Topology*, vol. 5, no. 3, pp. 1–33, 2021.

[40] P. S. Heckbert and M. Garland, "Survey of polygonal surface simplification algorithms," 1997.

[41] U. Fugacci, M. Kerber, and H. Manet, "Topology-Preserving Terrain Simplification," in *Proceedings of the 28th International Conference on Advances in Geographic Information Systems*, 2020, pp. 36–47.

[42] M. Johnson and J.-H. Jung, "Instability of the betti sequence for persistent homology and a stabilized version of the betti sequence," *arXiv preprint arXiv:2109.09218*, 2021.

[43] E. N. Epperly, J. A. Tropp, and R. J. Webber, "Xtrace: Making the most of every sample in stochastic trace estimation," *SIAM Journal on Matrix Analysis and Applications*, vol. 45, no. 1, pp. 1–23, 2024.

[44] L.-H. Lim, "Hodge Laplacians on graphs," *Siam Review*, vol. 62, no. 3, pp. 685–715, 2020.

[45] F. R. Chung, *Spectral graph theory*, vol. 92. American Mathematical Soc., 1997.

[46] T. E. Goldberg, "Combinatorial Laplacians of simplicial complexes," *Senior Thesis, Bard College*, vol. 6, 2002.

[47] F. C. Botelho, Y. Kohayakawa, and N. Ziviani, "A practical minimal perfect hashing method," in *International Workshop on Experimental and Efficient Algorithms*, 2005, pp. 488–500.

6 Appendix

6.1 Proofs

Proof of Lemma 1 [17]: The Pairing Uniqueness Lemma [1] 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:

$$\text{low}_R[j] = i \Leftrightarrow r_R(i,j) \neq 0 \Leftrightarrow r_\partial(i,j) \neq 0 \tag{47}$$

Extending this result to (7) 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: 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 (10) as:

$$\beta_p^{i,j} = \dim\big(C_p(K_i)\big)) - \dim\big(B_{p-1}(K_i)\big) - \dim\big(Z_p(K_i) \cap B_p\big(K_j\big)\big) \tag{48}$$

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:

$$\beta_p^{i,j} = \text{rank}\big(I_p^{1,i}\big) - \text{rank}\big(\partial_p^{1,i}\big) - \dim\big(Z_p(K_i) \cap B_p\big(K_j\big)\big) \tag{49}$$

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

$$\dim\big(Z_p(K_i) \cap B_p(K_i)\big) = \big|\Gamma_p^{i,j}\big| \tag{50}$$

where $\Gamma_p^{i,j} = \Big\{ \text{col}_{R_{p+1}[k]} \neq 0 \mid 1 \leq \text{low}_{R_{p+1}}[k] \leq i \Big\}$. 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:

$$\bar{\Gamma}_p^{i,j} = \dim\big(B_p\big(K_j\big)\big) - \big|\Gamma_p^{i,j}\big| = \text{rank}\Big(R_{p+1}^{i+1,j}\Big) \tag{51}$$

Combining equations (50) with (51) yields:

$$\dim\Big(Z_{p(K_i)} \cap B_{p(K_j)}\Big) = \big|\Gamma_{p(i,j)}\big| = \dim\big(B_p\big(K_j\big)\big) - |\bar{\Gamma}_p^{i,j}| = \text{rank}\Big(R_{p+1}^{1,j}\Big) - \text{rank}\Big(R_{p+1}^{i+1,j}\Big) \tag{52}$$

Observing the final matrices in (52) are *lower-left* submatrices of $R_{\{p+1\}}$, thus the final expression (8) follows by applying Lemma 1 repeatedly. ■

6.2 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 (18). 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 [44]:

$$\tilde{H}_p(K;\mathbb{R}) \simeq \ker\big(\Delta_p(K)\big), \quad \beta_p = \text{nullity}\big(\Delta_p(K)\big) \tag{53}$$

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 [24]). In fact, as pointed out by [24], 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:

$$\Lambda\big(\Delta_p(K)\big) \doteq \Lambda\big(L_p^{\text{up}}\big) \uplus \Lambda\big(L_p^{\text{dn}}\big), \qquad \Lambda\big(L_p^{\text{up}}\big) \doteq \Lambda\big(L_{p+1}^{\text{dn}}\big) \tag{54}$$

where $A \doteq B$ and $A \uplus B$ 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.

6.3 Laplacian matvec

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 $D = \operatorname{diag}(\{\deg(v_i)\})$ 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:

$$L = D - A = \partial_1 \circ \partial_1^T, \qquad L[i,j] = \begin{cases} \deg(v_i) & \text{if } \; i = j \\ -1 & \text{if } \; i \sim j \\ 0 & \text{if } \; i \nsim j \end{cases} \tag{55}$$

By using the adjacency relation $i \sim j$ as in [45], the linear and quadratic forms of L may be succinctly expressed as:

$$L(x)_i = \deg(v_i) \cdot x_i - \sum_{i \sim j} x_j, \quad x^T L x = \sum_{i \sim j} \big(x_i - x_j\big)^2 \tag{56}$$

If G has m edges and n vertices taking labels in the set $[n]$, observe computing the product from (56) 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 for $p > 0$, we first require a generalization of the connected relation $i \sim j$ from (56). Denote with $\operatorname{co}(\tau) = \{\, \sigma \in K^{p+1} \mid \tau \subset \sigma \,\}$ the set of proper cofaces of $\tau \in K^p$, or *cofacets*, and the (weighted) *degree* of $\tau \in K^p$ with:

$$\deg_w(\tau) = \sum_{\sigma \in \operatorname{co}(\tau)} w(\sigma) \tag{57}$$

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 σ is in fact *unique* in the sense that $\{\sigma\} = \operatorname{co}(\tau) \cap \operatorname{co}(\tau')$ (see [46] for more details). Thus, we may use a relation $\tau \overset{\sigma}{\sim} \tau'$ to rewrite L_p^{up} element-wise:

$$L_p^{\text{up}}(\tau, \tau') = \begin{cases} \deg_w(\tau) \cdot w^{+}(\tau) & \text{if } \; \tau = \tau' \\ s_{\tau,\tau'} \cdot w^{+/2}(\tau) \cdot w(\sigma) \cdot w^{+/2}(\tau') & \text{if } \; \tau \overset{\sigma}{\sim} \tau' \\ 0 & \text{otherwise} \end{cases} \tag{58}$$

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

$$\left(L_p^{\mathrm{up}}\, x\right)_i = \deg_w(\tau_i) \cdot w^+(\tau_i) \cdot x_i + w^{+/2}(\tau_i) \sum_{\tau_j \overset{\sigma}{\sim} \tau_i} s_{\tau_i,\tau_j} \cdot x_j \cdot w(\sigma) \cdot w^{+/2}\left(\tau_j\right) \tag{59}$$

Observe $v \to L_p^{\mathrm{up}} v$ can be evaluated now via a very similar two-pass algorithm as described for the graph Laplacian by simply enumerating the boundary chains of the simplices of K^{p+1}.

Below is pseudocode showing how to evaluate a weighted (up) Laplacian matrix-vector multiplication built from a simplicial complex K with $m = \left|K^{p+1}\right|$ and $n = |K^p|$ in $O(m)$ time when $m > n$, assuming p is considered a small constant. Key to the runtime of the linear runtime operation is the constant-time determination of orientation between p-faces ($s_{\tau,\tau'}$) and—for sparse complexes—the use of a deterministic $O(1)$ hash table $h : K^p \to [n]$ for efficiently determining the appropriate input/output offsets (i and j).

Require: Fixed oriented complex K of size $N = |K|$, $n = |K^p|$, $m = \left|K^{p+1}\right|$
Optional: Weight arrays $w_{p+1} \in \mathbb{R}_+^m$ and $w_p \in \mathbb{R}_+^n$
Output: $y = \left(W_p \circ \partial_{p+1} \circ W_{p+1} \circ \partial_{p+1}^T \circ W_p\right)x$

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

Algorithm 1: Matrix-free up-Laplacian matrix-vector multiplication algorithm.

In general, the signs of the coefficients $\mathrm{sgn}([\tau], \partial[\sigma])$ and $\mathrm{sgn}([\tau'], \partial[\sigma])$ depend on the position of τ, τ' as summands in $\partial[\sigma]$, 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 Algorithm 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 [46])—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 and p is a small constant, we can remove the storage requirement entirely and simply fix the sign pattern during the computation.

A subtle but important aspect of algorithmically evaluating (59) 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 Algorithm 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 PMHFs over fixed input sets of size n in $O(n)$ time and $O(n \log m)$ bits with deterministic $O(1)$ access time [47]. 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.

6.4 Choosing a weight function

Given (22), a natural question to ask whether there exists a weight function w that is more appropriate for rank computations. Since different weight choices yield different eigenvalue distribution in $\mathcal{L}_p$, the ideal choice is one that is the most amenable to compute. For the purpose of improving the condition number of the underlying Laplacian, consider the following optimization problem:

$$\begin{aligned} &\max_{w \in \mathbb{R}^n} \lambda_{\min}\big(\mathcal{L}_p(w)\big) \\ &\text{subject to } w > 0, \mathbf{1}^T w = 1 \end{aligned} \tag{60}$$

In the spectral graph theory, a specialization of this problem (for $p = 1$) arises under the guise of related problems, such as maximizing *algebraic connectivity* under a fixed total edge weight or finding the "fastest mixing time" of a Markov process. Fortunately, (60) is a convex optimization problem, which can be formulated as a semi-definite program (SDP) with variables $\gamma, \beta \in \mathbb{R}, w \in \mathbb{R}^n$ under positivity and sum-to-one constraints:

$$\begin{aligned} &\max_{\gamma \in \mathbb{R}} \gamma \\ &\text{subject to } \gamma I \preceq \partial_1 D_1(w) \partial_1^T + \beta \mathbf{1}\mathbf{1}^T, \quad w > 0, \quad \mathbf{1}^T w = 1 \end{aligned} \tag{61}$$

In general, we do not know if this approach generalizes for higher-order Laplacians, but in practice we found the weight function w^* that optimizes (60) to indeed be the easiest to approximate.