STABLE HOMOLOGY-BASED CYCLE CENTRALITY MEASURES

JOHN RICK D. MANZANARES* AND PAUL SAMUEL P. IGNACIO

ABSTRACT. Network centrality measures play a crucial role in understanding graph structures, assessing the importance of nodes, paths, or cycles based on directed or reciprocal interactions encoded by vertices and edges. Estrada and Ross extended these measures to simplicial complexes to account for higher-order connections. In this work, we introduce novel centrality measures by leveraging algebraically-computable topological signatures of cycles and their homological persistence. We apply tools from algebraic topology to extract multiscale signatures within cycle spaces of weighted graphs, tracking homology generators persisting across a weight-induced filtration of simplicial complexes built over point clouds. This approach incorporates persistent signatures and merge information of homology classes along the filtration, quantifying cycle importance not only by geometric and topological significance but also by homological influence on other cycles. We demonstrate the stability of these measures under small perturbations using an appropriate metric to ensure robustness in practical applications. Finally, we apply these measures to fractal-like point clouds, revealing their capability to detect information consistent with, and possibly overlooked by, common topological summaries.

1. INTRODUCTION

Many complex networks, such as social networks [socnet] and telecommunication networks [telnet] use graph-based centrality measures to determine the relative significance of nodes or cycles in the network. The derivations of the measures of central tendency in Statistics reflect the idea that a single value can represent the entire distribution of a data set. In particular, the mode of data sets is comparable to the degree centrality of graphs in that it projects importance through frequencies. Similarly, closeness centrality is akin to the median of data in the sense that it identifies nodes that are reachable via short paths from any node as "central" nodes.

Giscard and Wilson [1] introduced the loop-centrality measure that uses the number of walks that intersect a loop to measure its importance. They found that this centrality measure has the ability to detect essential proteins in protein-protein interaction networks. For the same purpose, Estrada and Ross [2] explored an extension of this centrality measure to finite-dimensional substructures where higher-order connectivity networks (such as co-author [coauthor] and social contact [soccont] networks) are represented by triangles and tetrahedra. A widely accepted notion of importance for cycles in a simplicial complex regards long-lived cycles as essential features of data, while short-lived cycles that appear are likely to be due to noise or sampling errors. However, Bubenik et al. [3] demonstrated that short-lived

cycles hold important information that can be used to estimate the curvature of surfaces.

In this study, we propose novel centrality measures that leverage the persistence of homology classes and their merge history along the filtration. Integral to this is the development of an algorithm that captures the merge history of homology classes. These homology-based centrality measures produce, for all cycle generators, curves that reflect the relative importance of the corresponding generator throughout its entire evolution. By applying these centrality measures on toy models, we demonstrate the consistency of detected information by these measures to other topological summaries, and highlight its ability to capture new information possibly missed by other summaries. Accordingly, we establish some properties that include the stability of these measures under a distance analogous to norms in Lebesgue spaces and persistence landscapes.

2. Preliminaries

This section lays the groundwork for extracting multiscale topological signatures from weighted graphs. First, we model higher-order interactions among vertices in the graph using simplicial complexes, similar to the approach used in loop centrality [2]. This captures interactions beyond simple pairwise connections. Next, we refine this model by applying a weight-based filtration. Filtration progressively removes less important edges, resulting in a nested sequence of increasingly coarser simplicial complexes. Finally, we leverage the concept of simplicial homology to analyze these filtered complexes. Simplicial homology allows us to identify and track cycle generators, which are loops within the complex that cannot be continuously deformed into a point. By focusing on these generators at each stage of the filtration, we can build a multi-scale record of the graph's overall topological structure. This is crucial to designing our cycle centrality measures later. Throughout this chapter, we shall use [**ctop**] and [**ten**] as standard references for discussions involving simplicial and persistent homology.

2.1. **Simplicial Complex.** We dive into the concept of simplicial homology and its application in modeling graphs. Although graphs represent connections between pairs of nodes (edges), simplicial complexes offer a more versatile framework. They allow us to capture not only pairwise interactions but also higher-order relationships between multiple nodes using simplices of various dimensions. This ability to encode complex interactions makes them particularly suitable for analyzing graphs with intricate connections beyond simple edges.

Definition 2.2. An *(abstract) simplicial complex* is a collection $\mathscr{C}$ of subsets of a finite set V such that $\tau \subseteq \sigma \in \mathscr{C}$ implies $\tau \in \mathscr{C}$. An element $\sigma \in \mathscr{C}$ is called an *(abstract) simplex* with *dimension* $|\sigma| - 1$, and the largest such dimension among all simplices in $\mathscr{C}$ defines the dimension of $\mathscr{C}$.

In the Euclidean space, lower-dimensional simplices are named as *vertex*, *edge*, *triangle*, and *tetrahedron* for a 0-simplex, a 1-simplex, a 2-simplex, and a 3-simplex, respectively. Furthermore, the higher-dimensional simplices are polytopes analogous to triangles and tetrahedra.

We can construct a simplicial complex based on a metric or a weight function. A metric-based example is the *Vietoris-Rips complex* or the *Rips complex*.

Definition 2.3. Let $\epsilon > 0$. Suppose that (M, d) is a metric space and $P = \{p_i\}$ is a finite subset of M. The *Vietoris-Rips complex* $\mathscr{R}_\epsilon(P)$ of P, with threshold ϵ, is a simplicial complex whose k-simplices correspond to $(k + 1)$-tuples of points in P such that $d(p_i, p_j) \leq 2\epsilon$ for any pair of integers (i, j).

Rips complexes form an n-simplex when every pair from the $n + 1$ points is connected by a 1-simplex or an edge. Figure 1 illustrates a Rips complex of a point cloud with six initial points from the Euclidean space.
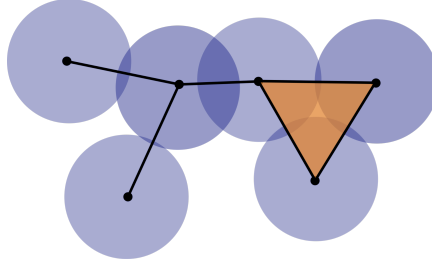

FIGURE 1. Rips complex of a point cloud

The simplices of an abstract simplicial complex provide the fundamental building blocks for constructing chains in chain space. Each k-simplex acts as a single unit within a k-chain. By combining these simplices with coefficients from $\mathbb{Z}/2\mathbb{Z}$ (which essentially encode the presence or absence of edges), we can create formal sums that represent more intricate relationships within the complex.

Definition 2.4. A *k-chain* is a formal sum of k-simplices with coefficients coming from the field $\mathbb{Z}/2\mathbb{Z}$. The *chain space*, denoted by C_k, is the free Abelian group generated by all possible k-simplices in the complex.

These k-chains serve as the building blocks for studying the topological properties of the simplicial complex. However, not all k-chains represent actual cycles within the complex. To distinguish cycles from other chains, we introduce the concept of a boundary operator.

Definition 2.5. The *boundary operator* $\partial_k : C_k \to C_{k-1}$ is the linear extension defined on the generators (individual k-simplices) given by

$$\partial_k \left([x_0, \ldots, x_j, \ldots, x_n]\right) = \sum_{i=0}^{n} [x_0, \ldots, \hat{x_i}, \ldots, x_n]$$

where $\hat{x_i}$ indicates that x_i is omitted.

Mathematically, the boundary operator acts on a k-simplex by summing over all its $(k - 1)$-dimensional faces. For each face, it creates a new $(k - 1)$-dimensional chain with the opposite coefficient. Intuitively, this captures the idea that the boundary of a k-dimensional object is made up of its $(k - 1)$-dimensional faces.

Definition 2.6. The *cycle space* Z_k and *boundary space* B_k of C_k are defined by

$$Z_k = \ker \partial_k, \text{ and } B_k = Im \ \partial_{k+1}.$$

We refer to elements of Z_k as k-cycles and those of B_k as k-boundaries.

Intuitively, these chains in the cycle space represent closed loops (k-cycles) within the complex that cannot be continuously deformed into a single point. Meanwhile, the chains in the boundary space represent the edges or borders of higher-dimensional simplices.

Applying the boundary operator twice, we confirm that $\partial_k \circ \partial_{k+1} = 0$. This property implies that every element of the boundary space is also a cycle.

2.7. **Simplicial Homology.** Now that we can distinguish cycles and boundaries, we can introduce the concept of homology, which focuses on the essence of cycles, capturing their topological properties rather than their specific form.

Definition 2.8. Let $\mathscr{C}$ be a simplicial complex. The *kth homology group* of $\mathscr{C}$, written as $H_k(\mathscr{C})$, is the quotient group Z_k/B_k. Two cycles in $Z_k(\mathscr{C})$ lying in the same homology class in this space are said to be *homologous*. We refer to the rank of $H_k(\mathscr{C})$ as the *kth Betti number* of $\mathscr{C}$.

In other words, the homology group is formed by taking the cycle space and "glue" any cycles that differ only by a boundary. This captures the intrinsic topological features of the complex at dimension k, independent of the specific choices of representatives for these cycles. The Betti number provides a numerical measure of the number of independent holes of dimension k in the complex.

By modding out the cycle space by the boundary space, we effectively exclude cycles that are simply the boundaries of higher-dimensional simplices. This ensures that we focus on nontrivial cycles that capture the true topological features of the complex. Lastly, homologous cycles represent k-dimensional loops within the complex that can be continuously deformed into each other.
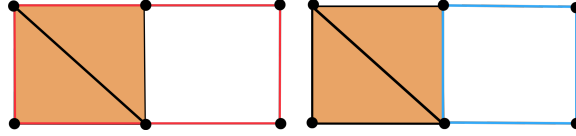

FIGURE 2. Two (red and blue) homologous 1-cycles

The latter remark can be further visualized as follows. Consider two 1-cycles colored red and blue in the simplicial complex shown in Figure 2. These cycles may appear different, but if their difference can be expressed as the boundary of a 2-simplex in the complex, they are considered homologous. This implies that these cycles can be continuously deformed, and any cycle may represent the homology class.

The generators of the homology group play a crucial role. They represent the distinct topological cycles embedded within our combinatorial model of the graph. Collectively, these cycles characterize the overall topology of the graph. Intriguingly, it is these generators and their corresponding homology classes that we leverage to define our centrality measures. However, a key point to remember is that equivalence within a homology class allows any generator to represent its class. For example, in Figure 2, either the red or blue 1-cycle can represent the same topological hole. This raises a natural question: How do we effectively choose representatives for these homology classes? This is where the concept of persistence comes into play, which we will explore in the next section.

2.9. **Filtrations and Persistent Homology.** This section explores filtrations and persistent homology, powerful tools for analyzing topological features within simplicial complexes. These tools allow us to track how the underlying topological structure of a complex evolves as we progressively remove or modify its simplices based on some criteria.

We begin by introducing the concept of weights in simplicial complexes. A weight function $w : \mathscr{C} \to \mathbb{R}_{\geq 0}$ assigns weights (nonnegative real numbers) to each simplex σ in the complex $\mathscr{C}$. This function allows us to differentiate between simplices and prioritize their removal during the filtration process. A common example of a weight function is the distance function, where the weight of a simplex might be defined as the maximum distance between any two points within that simplex. Higher weight values would then correspond to simplices with larger maximum distances, potentially representing less connected or more spread-out regions of the complex.

A filtration is a nested sequence of subcomplexes obtained by progressively removing simplices exceeding a certain weight threshold. Different weight order (e.g. $\geq$ order) leads to different filtration types. Intuitively, as the weight threshold increases, simplices with lower importance (based on the weight function) are removed. This reveals a coarser version of the original complex at each step, where only the most significant simplices remain.

This filtration process induces a sequence $\{\mathscr{C}_{w_i}\}$ of simplicial complexes. These complexes capture the connectivity structure of the original complex at each weight threshold. Persistent homology leverages this sequence to track the "persistence" of topological features (cycles) across different weight thresholds. In simpler terms, we are interested in how these features are born, die, or evolve as the filtration progresses.

Definition 2.10. Let $\{\mathscr{C}_{w_i}\}$ be a filtered simplicial complex induced from a filtration of a simplicial complex $\mathscr{C}$. The *kth $w_i - w_j$ persistent homology group* of $\mathscr{C}$, denoted by $H_k^{w_i, w_j}$, is the image of the induced homomorphism

$$H_k(\mathscr{C}_{w_i}) \xrightarrow{w_{*k}^{w_i, w_j}} H_k(\mathscr{C}_{w_j}).$$

This group captures the homology classes "born" at weight w_i and persisting (remaining in the homology group) until weight w_j. This group encodes how long homology classes survive under the filtration. For instance, a cycle that persists across a wide range of weight values suggests a more robust topological feature in the complex.

To facilitate future discussions, we introduce an abuse of notation where σ can represent both a simplex and the corresponding cycle it defines.

Definition 2.11. Suppose $\epsilon_0 < \epsilon < \epsilon'$. We say that a homology class $[\sigma]$ is *born* at $\mathscr{C}_\epsilon$ if $[\sigma] \in H_k^{\epsilon, \epsilon'}$ but $[\sigma] \notin H_k^{\epsilon_0, \epsilon'}$. On the contrary, we say that $[\sigma]$ born at $\mathscr{C}_{\epsilon_0}$ *dies* entering $\mathscr{C}_{\epsilon'}$ if $w_k^{\epsilon_0, \epsilon'}([\sigma]) = 0$ but $w_k^{\epsilon_0, \epsilon}([\sigma]) \neq 0$. Let $b(\sigma)$ and $d(\sigma)$, respectively, denote the birth and death of $[\sigma]$. We say that σ_i *gives birth to* the cycle $\sigma = \sum \sigma_i$ if $b(\sigma) = b(\sigma_i)$.

When two homology classes merge due to simplex removal, the elder rule comes into play. This rule selects the generator formed at the lower threshold as the natural representative of the merged class. A key observation is made: even when a different generator survives a merge, the persistence information can be "transferred" to

the surviving one. This highlights the dynamics of merging classes and motivates further investigation into how these classes evolve under the filtration.

Definition 2.12. Let σ and ν be k-cycles. We say that σ and ν are *k-near* if there exists a k-chain common to both σ and ν. Furthermore, the kth homology classes $[\sigma]$ and $[\nu]$ are *k-near* if σ and ν are k-near.

The notion of k-nearness formalizes the idea that two cycles are close in the complex.

We also define the concept of merging governed by the elder rule. This definition clarifies the direction of merging, where a class with an earlier death threshold merges with the longer-lived class.

Definition 2.13. Let $[\sigma]$ and $[\nu]$ be distinct kth homology classes in a simplicial complex $\mathscr{C}_\epsilon$ where $d(\sigma) \neq d(\nu)$. We say that $[\sigma]$ and $[\nu]$ *merge* at time $\epsilon' = \min\{d(\sigma), d(\nu)\}$ if

$$\sigma + \nu = \rho$$

for some k-boundary ρ.

Within the framework of persistent homology, the merging of homology classes due to simplex removal is governed by the well-established *elder rule*. This rule dictates that the generator formed at the lower filtration threshold becomes the designated representative for the newly formed merged class. Notably, the persistence information associated to the eliminated generator can still be "transferred" or "inherited" by the surviving one.

However, it is crucial to distinguish between the merging process and the act of gluing a boundary to a cycle. While merging signifies the disappearance of a topological feature upon simplex removal, gluing a boundary to a cycle essentially fills in the hole or closes the gap within the cycle. Although the resulting structure remains homologous, it falls outside the scope of current persistent homology methods. Addressing this distinction involves investigating the sequential gluing actions across the filtration.

The dynamic nature of merging classes is underscored by the observed phenomenon of information transfer and the need to resolve boundary gluing. These observations prompt further inquiry into the evolution of these classes under the filtration process.

2.14. **Algorithm for Class Merges.** The computation of persistent homology often involves the algorithm introduced by Zomorodian [5]. This algorithm identifies the representatives (cycles) for each homology class. It mainly involves reducing the kth boundary matrix constructed by arranging the $(k-1)$- and k-simplices in increasing births that comprise the rows and columns, and identifying the entries using the function

$$\partial_k[i,j] = \begin{cases} 1 & \text{if } \sigma_i \text{ is a codimension-1 face of } \sigma_j \\ 0 & \text{otherwise} \end{cases}.$$

The columns of the boundary matrix are ordered by birth.

A pseudocode of the algorithm [**ctop**] is presented in Algorithm 2.1, where $l(i)$ denotes the position of the lowest 1 in column i of the boundary matrix.

Let $c(\sigma)$ be the column associated with the simplex σ in the boundary matrix. We introduce the ordering $\sigma \prec \delta$ to mean that either $b(\sigma) < b(\delta)$, or if $b(\sigma) = b(\delta)$

Algorithm 2.1 Standard reduction algorithm for boundary matrices.

Require: Boundary matrix of order m
Ensure: Reduced boundary matrix
 1: **for** $j = 2$ to m **do**
 2:     **while** there exists $i < j$ with $l(i) = l(j)$ **do**
 3:         add column i to column j
 4:     **end while**
 5: **end for**

---

and $c(\sigma)$ appears first in the column of the boundary matrix. Additionally, let $r(\sigma)$ be the chain produced in $c(\sigma)$ by the reduction algorithm.

Lemma 2.15. *Suppose that $[\sigma]$ is a kth homology class in a filtered simplicial complex $\mathscr{C}$. If $[\sigma]$ is not k-near to any homology class $[\nu]$, then the reduction algorithm produces a unique class representative for $[\sigma]$.*

Proof. We assume that $\sigma = \sum_i \sigma_i$ and $\nu = \sum_j \nu_j$ are not k-near cycles where some σ_i and ν_j respectively give birth to σ and ν. Furthermore, let $\nu_j \prec \sigma_i$. Note that the kth homology class representative depends on the associated kth boundary matrix, which is only concerned with simplices of dimensions $k-1$ and k. Therefore, we can omit the case where σ and ν has a common simplex of dimension 0 to $k-2$.

Suppose that σ and ν have no shared $(k-1)$-simplex. Then, for every simplex δ, $l(\delta) \neq l(\sigma_i)$ for every integer $1 \leq i \leq n$. Therefore, the representative of $[\sigma]$ must be unique, since no simplex can be added with σ_i to form another representative.

Now, consider the case where σ and ν intersect at some $(k-1)$-faces. For the sake of contradiction, suppose that σ has a representative containing some simplices from ν. Since σ and ν has no shared k-simplex, $r[\sigma]$ must contain all simplices of ν. Otherwise, $r[\sigma]$ ceases to be a cycle if it does not contain all simplices of ν. Note that, in the reduction algorithm, $c(\nu_j)$ reduces to a zero vector before reduction happens on $c(\sigma_i)$. Hence, $c(\sigma_i)$ cannot contain the simplex ν_j of ν. Therefore, we arrive at a contradiction so $[\sigma]$ must have a unique class representative. $\square$

Consider two k-near cycles σ and ν that share a chain δ. The shared simplices, along with simplices from σ and ν that touch these shared simplices, can create situations where some simplices in $\sigma + \delta$ might have the same position of the lowest one as some simplices in $\nu + \delta$ and δ. We call such connection points as *junctures*.

Lemma 2.16. *Suppose that $[\sigma]$ and $[\nu]$ are distinct kth homology classes appearing in the filtration of a simplicial complex satisfying $\nu \prec \sigma$. Let $\nu = \sum_p \nu_p$ be a cycle representing $[\nu]$, where ν_p gives birth to ν. Additionally, assume σ and ν are k-near with common chain δ. If $\nu_p \in \delta$ and ν does not contain simplices of σ except δ, then the representative of $[\sigma]$ produced by the reduction algorithm is given by $\sigma + \nu$. Otherwise, if $\nu_p \notin \delta$ then the representative of $[\sigma]$ produced by the reduction algorithm is given by σ.*

Proof. Suppose that $\sigma = \sum_i \sigma_i$ and ν are k-near with the intersection δ, and $\nu_p \in \delta$. Let σ_i be the simplex giving birth to σ. Note that $r(\sigma_i)$ only depends on the junctures and the algorithm proceeds by having $\sigma + \delta$ in $r(\sigma_i)$. If $\delta \in r(\sigma)$, then we arrive at a contradiction since $\nu_p \in \delta$ and ν_p gives birth to ν so $c(\nu_p)$ is a zero vector.

If there exists no juncture where $\sigma_j \in \sigma + \delta$ and $\nu_q \in \nu$ such that $l(\sigma_j) = l(\nu_q)$, then $r(\sigma_i)$ ceases to be a cycle. Thus, there exists $\sigma_j \in \sigma + \delta$ and $\nu_q \in \nu$ such that $l(\sigma_j) = l(\nu_q)$. By the algorithm, $r(\nu_q) \in r(\sigma_i)$, and we have $\nu + \delta \in r(\sigma_i)$. Consequently,

$$r(\sigma_i) = (\sigma + \delta) + (\nu + \delta) = \sigma + \nu.$$

Therefore, we have proved the lemma. □

We are now able to provide a mechanism to track the merge information between homology classes and a criterion for when such happens.

Theorem 2.17. *Let $[\sigma]$ and $[\nu]$ be distinct persistent homology classes where $\sigma \prec \nu$. Then $[\sigma]$ merges with $[\nu]$ at $\epsilon = d(\sigma)$ if and only if $[\sigma]$ and $[\nu]$ are near.*

Proof. Suppose $[\sigma]$ merges with $[\nu]$. If $[\sigma]$ and $[\nu]$ are not near, then both classes have unique cycle representatives that are not near. Thus, if $\sigma + \nu$ is a boundary at ϵ then $d(\sigma) = d(\nu)$. This equation is a contradiction to the assumption that the classes merge. Therefore, $[\nu]$ and $[\sigma]$ must be near.

Conversely, suppose that $[\sigma]$ and $[\nu]$ are near. By Lemma 2.16, we can choose $\sigma + \nu$ as representative of $[\sigma]$. Observe that

$$(\sigma + \nu) + \nu = \sigma.$$

Since $\sigma \prec \nu$, σ must be a boundary at ϵ. Therefore, $[\sigma]$ merges with $[\nu]$. □

In addition to keeping track of the cycle representatives yielded by the reduction algorithm, we are also interested in integrating the *snowball effect* of successive merging classes in persistent homology. This necessitates the following definition.

Definition 2.18. Consider the induced homomorphism

$$H_k(\mathscr{C}_{w_i}) \xrightarrow{w_{*k}^{w_i,w_j}} H_k(\mathscr{C}_{w_j})$$

and let $[\sigma] \in H_k(\mathscr{C}_{w_j})$. We define the *first order merge cluster* of $[\sigma]$ at w_j as

$$M_1[\sigma, w_j] := (w_{*k}^{w_i,w_j})^{-1}([\sigma]),$$

that is, the set of homology classes merging with $[\sigma]$ at threshold w_j. Inductively, for every integer $n \geq 2$, the *nth order merge cluster* of $[\sigma]$ at w_j is defined as

$$M_n[\sigma, w_j] = \bigcup_{\tau \in M_{n-1}[\sigma, w_j]} M_1[\tau, w_j].$$

The first-order merge clusters refer to all other classes that merges with a specific class $[\sigma]$ across the filtration until the threshold w_j. Higher-order merge clusters build on this idea. Imagine a class that merges with another class, which itself merges with a third class. All these classes are considered part of the same merge cluster because their merges are ultimately connected. The definition captures this cascading effect by recursively defining higher-order merge clusters as the union of all first-order merge clusters of classes within the previous-order merge cluster.

Theorem 2.17 offers a way to determine if two classes merge at a specific threshold. However, it might involve checking many pairs of cycles representing the classes, which can be computationally expensive. This is because any homologous cycle can represent a class, leading to a large number of potential pairings to explore. Corollary 2.19 aims to reduce the number of pairings that need to be checked for merging. It leverages the concept of k-nearness between cycles representing classes.

Corollary 2.19. *Suppose that the homology class $[\sigma]$ merges with $[\nu]$. Let $[\delta]$ be a homology class such that $\sigma \prec \delta \prec \nu$. If $[\nu]$ and $[\delta]$ are near then either $[\delta]$ merges with $[\sigma]$, or there exists a homology class that merges with $[\nu]$ whose nth order merge cluster contains $[\delta]$ for some integer $n \geq 1$.*

Proof. Suppose $[\nu]$ and $[\delta]$ are near. If $[\delta]$ merges with $[\nu]$, then we have proved the corollary. Suppose $[\delta]$ merges with a homology class $[\lambda]$ where $\delta \prec \lambda \prec \nu$. Note that a representative of $[\lambda]$ includes some simplices of δ. Since $[\nu]$ and $[\delta]$ are near, $[\nu]$ and $[\lambda]$ must also be near. Therefore, $[\lambda]$ merges with $[\sigma]$. $\qquad\square$

We can implement Theorem 2.17 and Corollary 2.19 in an algorithm to obtain the first-order merge clusters of each homology class. Algorithm 2.2 presents a pseudocode for finding the first order merge clusters.

Algorithm 2.2 Identifying First-Order Merge Clusters from Class Representatives.

Require: Representatives $\{\sigma_i\}_{i=1}^{l}$ ordered by ascending birth followed by death
Ensure: Collection containing first order merge clusters of the homology classes
1: Initialize an array M of length l
2: **for** $i \in \{2, \ldots, l\}$ **do**
3:     **for** $j \in \{1, \ldots, i-1\} \setminus \bigcup_{k=1}^{i-1} M[k]$ **do**
4:         **if** $[\sigma_i]$ is k-near to $[\sigma_j]$ and $d(\sigma_i) \geq d(\sigma_j) > b(\sigma_i)$ **then**
5:             insert j to $M[i]$
6:         **end if**
7:     **end for**
8: **end for**

---

3. Cycle Centrality

This chapter explores the core of our study, defining centrality measures for cycles based on persistent homology. These measures go beyond a cycle's mere existence and capture its topological significance and influence within a network. We achieve this by considering two key aspects, namely persistence and merge dynamics. Our goal is to define centrality measures that are monotonic functions. This means that the importance of a cycle, as measured by our centrality, never decreases as the filtration progresses.

3.1. **Centrality Measures.** We propose three centrality functions, denoted by J_1, J_2, and J_3, to capture the evolving importance of cycles. Each function is defined for a specific homology class representative $[\sigma]$ and a filtration threshold ϵ. All the functions share a common structure. They possess a base value of 0 before the birth threshold of $[\sigma]$, and exhibit a piece-wise linear behavior after birth, reflecting changes in importance.

The first centrality measure J_1 aims to capture and quantify the total persistence accumulated by $[\sigma]$ and all classes that directly merge with it up until the threshold ϵ. Hence, if $P_\epsilon(\sigma)$ is the persistence of σ at ϵ, the first centrality function has the form

$$J_1(\sigma, \epsilon) = \begin{cases} 0 & \text{for } \epsilon \le b(\sigma) \\ P_\epsilon(\sigma) + \displaystyle\sum_{[\varsigma] \in M_1[\sigma, \epsilon]} P_\epsilon(\varsigma) & \text{for } \epsilon > b(\sigma) \end{cases} \cdot$$

As each $P_\epsilon(\sigma)$ is monotonic and stabilizes when $\epsilon > d(\sigma)$, this function is piece-wise linear and monotonic. It captures the cascading simple aggregate of persistence pooled from cycle representatives that altogether merge directly to an older cycle. We view this as the homological importance of cycles — if many cycles merge to an old cycle, then its homological significance is proportionally increased.

One caveat of the function above is that it treats all merging instances similarly regardless of when the merge happens along the filtration. Hence, we also consider a second centrality function J_2 by refining J_1 to account for the time when instances of merging happen. It introduces a scaling function that assigns a weight to the persistence of each merging class. This allows us to prioritize either early or late merges in the centrality calculation. We can write this function as

$$J_2(\sigma, \epsilon) = \begin{cases} 0 & \text{for } \epsilon \le b(\sigma) \\ P_\epsilon(\sigma) + \displaystyle\sum_{[\varsigma] \in M_1[\sigma, \epsilon]} f_\sigma(\varsigma) P_\epsilon(\varsigma) & \text{for } \epsilon > b(\sigma) \end{cases} \cdot$$

When early merges are considered more influential, we can define $f_\sigma(\varsigma)$ as $d(\varsigma)/d(\sigma)$. Conversely, when late merges are considered more important, we can define $f_\sigma(\varsigma)$ as $1 - d(\varsigma)/d(\sigma)$.

We can also generalize the cascading effect of merging by considering the indirect transfer of persistence from merging instances prior to a given merge time. This is equivalent to modifying the centrality function to account for higher-order merging clusters, and yields a third centrality function J_3 given by

$$J_3(\sigma, \epsilon) = \begin{cases} 0 & \text{for } \epsilon \le b(\sigma) \\ P_\epsilon(\sigma) + \displaystyle\sum_{r} \sum_{[\varsigma] \in M_r[\sigma, \epsilon]} f_\sigma(\varsigma) P_\epsilon(\varsigma) & \text{for } \epsilon > b(\sigma) \end{cases} \cdot$$

Here, we allow the definition $f_\sigma(\varsigma) = 1$ to generalize the function J_1.

In general, our centrality measures are of the form $J_n : \Lambda \times W \to \mathbb{R}_+$ where Λ is the collection of all non-trivial persistent homology classes from a filtration. Hence, each persistence diagram produces a family of centrality functions. We can visualize the time-evolving centrality function of each homology class by constructing a piecewise-linear plot where sudden increases represent the effect of merging. We call these plots the J_n *centrality plots* of dimension k, where k refers to the dimension of the homology classes we are concerned with. We can also create heat maps from contiguous heat bars each capturing the monotonic growth of centrality as captured by our measures. We illustrate a centrality plot in Figure 3 and omit the heat map representation.

3.2. **Stability of Centrality Measures.** This subsection investigates the stability of the centrality measures we defined earlier. Stability ensures that small changes in the network data, such as slight adjustments to edge weights, will not lead to drastic changes in the calculated centrality of cycles.

First, we establish that all three centrality measures are monotonic. Intuitively, as we explore the network at a coarser scale, a hole might encounter more merging
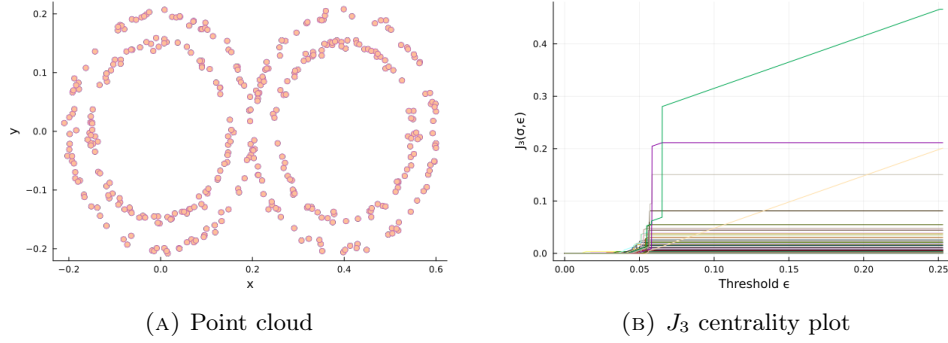
(A) Point cloud

(B) J_3 centrality plot

FIGURE 3. The J_3 centrality plot, with $f_\sigma = 1$, of dimension 1 produced by the Rips filtration of the point cloud sampled around a wedge sum of two annuli.

partners, potentially accumulating more persistence and thus increasing its centrality. In the following lemma, we denote $J_n(\sigma, \epsilon)$ by $J_{n,\sigma}(\epsilon)$.

Lemma 3.3 (Monotonicity). *If $\epsilon < \epsilon'$, then $J_{n,\sigma}(\epsilon) \leq J_{n,\sigma}(\epsilon')$.*

Proof. It follows from definition that $M_k[\sigma, \epsilon] \subseteq M_k[\sigma, \epsilon']$ for $\epsilon < \epsilon'$ and $k \geq 1$. Since P_ϵ is monotonic and f is constant with respect to ϵ, the conclusion follows. □

To analyze stability, we introduce the concept of a p-centrality norm. This norm assigns a numerical value to each hole's centrality, capturing the overall importance of that hole. The specific value of p influences how the norm prioritizes different aspects of the centrality function. For example, $p = 1$ emphasizes the average centrality, while $p = \infty$ focuses on the maximum centrality value achieved by the hole.

Definition 3.4. Let $\mathcal{J}_n = \{J_{n,\sigma} | [\sigma] \in \Lambda\}$ denote the collection of centrality functions generated by the set of persistent homology classes Λ. The *p-centrality norm* given by

$$\|J_{n,\sigma}(d^*)\|_p = \begin{cases} \left(\int_0^{d^*} (J_{n,\sigma}(x))^p \, dx\right)^{1/p} & \text{if } 1 \leq p < \infty \\ J_{n,\sigma}(d^* & \text{if } p = \infty \end{cases}$$

where d^* is the minimum between $d(\sigma)$, and the largest geodesic distance between any two vertices in the largest cycle in $[\sigma]$.

Next, we define a metric called the p-centrality distance to compare collections of centrality norms. Since a natural order ceases to exist for the centrality measures, we appeal to a bottleneck-like distance approach. The computation considers optimal pairings between centrality norms in the two collections and focuses on the maximum difference between any two paired values. For computational efficiency, we match the condition in the bottleneck distance implementation in [6]. In this case, let $\Omega = 0 \times \{\|J_{n,\sigma}(d^*)\|_p^p : J_{n,\sigma} \in \mathcal{J}_n\}$ and $\Omega' = 0 \times \{\|J_{n,\sigma'}(d^{*\prime})\|_p^p : J_{n,\sigma'} \in \mathcal{J}_n'\}$. For $x_\sigma \in \Omega$, we define $\delta_{x_\sigma} = \|J_{n,\sigma}\|_p^p$. For a bijection $\phi : \Omega \cup \Delta \to \Omega' \cup \Delta$, define

$$(1) \qquad \|x_\sigma - \phi(x_\sigma)\|_\infty = \begin{cases} \frac{1}{2}\max\{\delta_{x_\sigma}, \delta_{\phi(x_\sigma)}\} & \text{if } \phi(x_\sigma) \in \Delta \\ |\delta_x - \delta_{\phi(x)}| & \text{otherwise} \end{cases}.$$

Definition 3.5. For $1 \leq p < \infty$, the *p-centrality distance* is given by

$$C_p \left(\mathcal{J}_n, \mathcal{J}'_n \right) = \inf_{\phi} \sup_{x \in X} \| x - \phi(x) \|_\infty$$

where the infimum is taken over all bijections from $\Omega \cup \Delta$ to $\Omega' \cup \Delta$.

For the case where $p = \infty$, we propose a distance akin to p-landscape distance [7, p. 94]. Note that we can order the centrality function $J_{n,\sigma}(d*)$ of each homology class $[\sigma]$ based on the maximum centrality values. Thus, we obtain a decreasing sequence $\{ \| J_{n,m} \| \}_m$ where m is a positive integer. The *p-centrality distance* is then given by

$$C_p \left(\mathcal{J}_{n,k}, \mathcal{J}'_{n,k} \right) = \sum_m \left\| J_{n,m} - J'_{n,m} \right\|_p .$$

Example 3.6. We examine how our proposed centrality measures behave with respect to perturbations of the point cloud in Figure 3a introduced by replacing each point (x, y) with $(x + \kappa_1, y + \kappa_2)$ for some $\kappa_1, \kappa_2 \in [-\kappa, \kappa]$. We then compute the 1-centrality distance between the centrality measures of the original point cloud and its perturbation, and replicate thirty simulations of this process to generate a distribution of 1-centrality distances represented by boxplots. In Figures 4, we observe how the distribution of 1-centrality distances varies across increasing levels of perturbations.
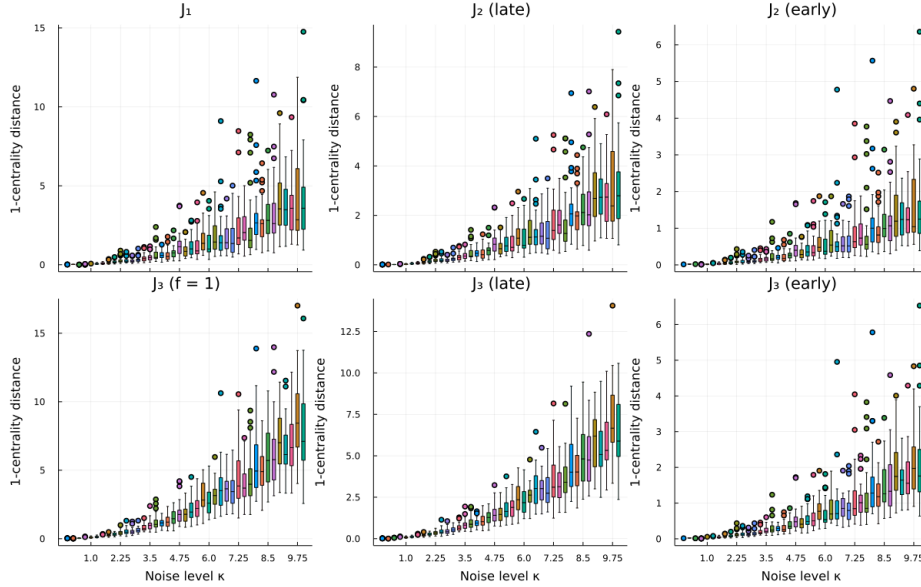

FIGURE 4. Boxplots for the 1-centrality distance between the centrality functions of the point cloud in Figure 3a and its perturbations for all noise levels.

We now establish bounds for the p-centrality distance to quantify the stability of the centrality measures. We simplify notations by dropping n and write J_σ when we consider any of the centrality functions $J_{1,\sigma}$, $J_{2,\sigma}$, and $J_{3,\sigma}$.

The following results focus on collections Λ and Λ' of persistent homology class representatives obtained through a filtration of a simplicial complex. We define K as the maximum persistence observed across all cycles in both Λ and Λ'. Additionally, we define q as the maximum number of cycles between Λ and Λ'. This expression represents the larger collection size for comparison purposes. Lastly, we assume that there exists a homology class with a non-empty first-order merge cluster.

Lemma 3.7. *Let $\mathcal{J}(\Lambda)$ be the collection of centrality measures generated from Λ. Then*

$$C_p\left(\mathcal{J}(\Lambda), \mathcal{J}\left(\Lambda'\right)\right) \leq \begin{cases} K^{1+\frac{1}{p}}\left(1+q\right) & \text{if } 1 \leq p < \infty \\ Kq(1+q) & \text{if } p = \infty \end{cases}.$$

Proof. Monotonicity of the centrality measures (Proposition 3.3) yields

$$J_\sigma(\epsilon) \leq P_\epsilon(\sigma) + \sum_r \sum_{[\nu] \in M_r[\sigma, d(\sigma)]} f_\sigma(\varsigma) P_\epsilon(\nu).$$

By the definitions of K and q, we have $J_\sigma(\epsilon) \leq K + f_\sigma(\varsigma) q K$. Thus, by the monotonicity of Lebesgue integrals, we obtain $\|J_\sigma(\epsilon)\|_p \leq \sqrt[p]{d^*}\left[K + f_\sigma(\varsigma) q K\right]$ where the integral is taken from $[0, d^*]$. Recall that $f_\sigma(\varsigma) \in (0, 1]$. Thus,

$$\|J_\sigma(\epsilon)\|_p \leq K^{1+\frac{1}{p}}(1+q). \tag{2}$$

Now, we define $X = \{\|J_\sigma\| : J_\sigma \in \mathcal{J}(\Lambda)\}$ and $Y = \{\|J'_\delta\| : J'_\delta \in \mathcal{J}(\Lambda')\}$. Assuming $|\Lambda| \leq |\Lambda'|$, we consider a bijection $\phi : X \cup \Delta \to Y$. Note that $|\|J_\sigma\|_p - \|J_\delta\|_p| \leq \max\{\|J_\sigma\|_p, \|J_\delta\|_p\}$. Using Equation 2, for any $x \in X$ and $y \in Y$,

$$|x - y| \leq K^{1+\frac{1}{p}}(1+q).$$

Now, suppose $x \in \Delta$. By Equation 1, we obtain

$$\frac{1}{2}\max\{x, \phi(x)\} \leq \frac{1}{2}K^{1+\frac{1}{p}}(1+q) < K^{1+\frac{1}{p}}(1+q).$$

Therefore, $C_p\left(\mathcal{J}(\Lambda), \mathcal{J}\left(\Lambda'\right)\right) \leq K^{1+\frac{1}{p}}(1+q)$ for $1 \leq p < \infty$.

We consider the case where $p = \infty$. Note that, for any $\epsilon \geq 0$, $J_\sigma(\epsilon) \geq 0$. Hence,

$$\sum_m \|J_m - J'_m\|_\infty \leq \sum_m \max\{J_m(d*), J_m(d*)\}.$$

Since $J_\sigma(\epsilon) \leq K + f_\sigma(\varsigma) q K$ and $f_\sigma(\varsigma) \in (0, 1]$, we have

$$\sum_m \|J_m - J'_m\|_\infty \leq \sum_m K(1+q).$$

By applying the definition of q, we have proven the lemma. $\square$

The value of K suggests a potential for variations in the lifetime of topological features. In the previous lemma, a smaller value of K contributes to tighter bounds.

We now introduce a constant that will be instrumental in establishing an upper bound for the centrality distance. This constant is given by

$$R(p) := \begin{cases} \sqrt[p]{2}K(1+q) & \text{if } 1 \leq p < \infty \\ 2q(1+q) & \text{if } p = \infty \end{cases}.$$

The next step is to demonstrate that the p-centrality distance between any two collections of centrality measures is always upper bounded by this constant and the bottleneck distance between the corresponding collections.

Theorem 3.8. *Let D and D' represent the persistence diagrams corresponding to the collections Λ and Λ'. Then*

$$C_p\left(\mathcal{J}(\Lambda), \mathcal{J}\left(\Lambda'\right)\right) \leq \begin{cases} R(p) \sqrt[p]{d_B(D, D')} & \text{if } 1 \leq p < \infty \\ R(p) d_B(D, D') & \text{if } p = \infty \end{cases}.$$

Proof. For any pair of homology classes $[\sigma]$ and $[\delta]$, application of the triangle inequality yields

$$|P_\epsilon(\sigma) - P_\epsilon(\delta)| \leq |d(\sigma) - d(\delta)| + |b(\sigma) - b(\delta)|.$$

By the definition of the bottleneck distance, we have $|P_\epsilon(\sigma) - P_\epsilon(\delta)| \leq 2d_B(D, D')$. For the left-hand expression, the maximum taken over all homology classes $[\sigma]$ and $[\delta]$ is $P_\epsilon(\sigma)$ or $P_\epsilon(\delta)$. Consequently, $K \leq 2d_B(D, D')$. The conclusion follows from Lemma 3.7. $\qquad\square$

The following corollaries leverage the combinatorial stability theorem [8, p. 123] to reformulate the previously established bounds in terms of the constant $R(p)$ and properties of the filtration functions. We omit the proofs for these corollaries as they directly apply the referenced theorem.

Corollary 3.9. *Let $w, w' : \mathscr{C} \to \mathbb{R}$ be monotone real-valued functions that filter the simplicial complex $\mathscr{C}$. Then*

$$C_p\left(\mathcal{J}(\Lambda), \mathcal{J}\left(\Lambda'\right)\right) \leq \begin{cases} R(p) \sqrt[p]{\|w - w'\|_\infty} & \text{if } 1 \leq p < \infty \\ R(p)\|w - w'\|_\infty & \text{if } p = \infty \end{cases}.$$

In the next corollary, we introduce another constant given by

$$R'(p) := \begin{cases} \sqrt[p]{2}K(1 + q') & \text{if } 1 \leq p < \infty \\ 2q'(1 + q') & \text{if } p = \infty \end{cases}$$

where $q' := \max\left\{\sum_r |M_r[\sigma, d(\sigma)]| : \sigma \in \Lambda \cup \Lambda' \text{ and } P_\epsilon(\sigma) \neq 0\right\}$.

The term q' represents the maximum number of successive merging of homology classes across Λ and Λ'.

Corollary 3.10. *Let D and D' be persistence diagrams corresponding to the collections Λ and Λ' and let Then*

$$C_p\left(\mathcal{J}_n(\Lambda), \mathcal{J}_n\left(\Lambda'\right)\right) \leq \begin{cases} R'(p) \sqrt[p]{d_B(D, D')} & \text{if } 1 \leq p < \infty \\ R'(p) d_B(D, D') & \text{if } p = \infty \end{cases}.$$

The previous corollary refines the bounds further by incorporating information about the merge clusters of homology classes. A smaller q' indicates less variation in how homology classes merge between Λ and Λ', leading to potentially tighter bounds.

Theorem 3.11 (Stability). *Let $w, w' : \mathscr{C} \to \mathbb{R}$ be monotone real-valued functions that filter the simplicial complex $\mathscr{C}$. The inequality*

$$C_p\left(\mathcal{J}_n(\Lambda), \mathcal{J}_n\left(\Lambda'\right)\right) \leq R'(p)\|w - w'\|_\infty.$$

holds when either $p = \infty$, or $1 \leq p < \infty$ and $q' > \left(1/\sqrt[p]{2}K\right) - 1$.

Proof. Suppose that $1 \leq p < \infty$ and $q' > \left(1/\sqrt[p]{2K}\right) - 1$. It follows from the combinatorial stability theorem and Corollary 3.10 that

$$\frac{1}{R'(p)}C_p\left(\mathcal{J}_n(\Lambda), \mathcal{J}_n\left(\Lambda'\right)\right) \leq \|w - w'\|_\infty^{\frac{1}{p}}.$$

The conclusion follows because $x^{\frac{1}{p}} \leq x$ for any $x \geq 0$.

Suppose $p = \infty$. The inequality $2q'(1 + q') \geq 1$ holds since $q' \geq 1$. Applying the combinatorial stability theorem, we obtain

$$\frac{1}{R'(p)}C_p\left(\mathcal{J}_n(\Lambda), \mathcal{J}_n\left(\Lambda'\right)\right) \leq \|w - w'\|_\infty.$$

$\square$

To evaluate the effectiveness of the bounds established in Corollary 3.10, we perform the following analysis. We consider the 1-centrality distances (refer to Example 3.6) calculated for various perturbation levels. From the bounds provided by Corollary 3.10, we subtract the actual 1-centrality distances. These differences are then plotted as boxplots across increasing levels of perturbations, similar to the approach used in the previous figures. This visualization allows us to assess how closely the theoretical bounds align with the empirical observations of the 1-centrality distance under increasing network perturbations.
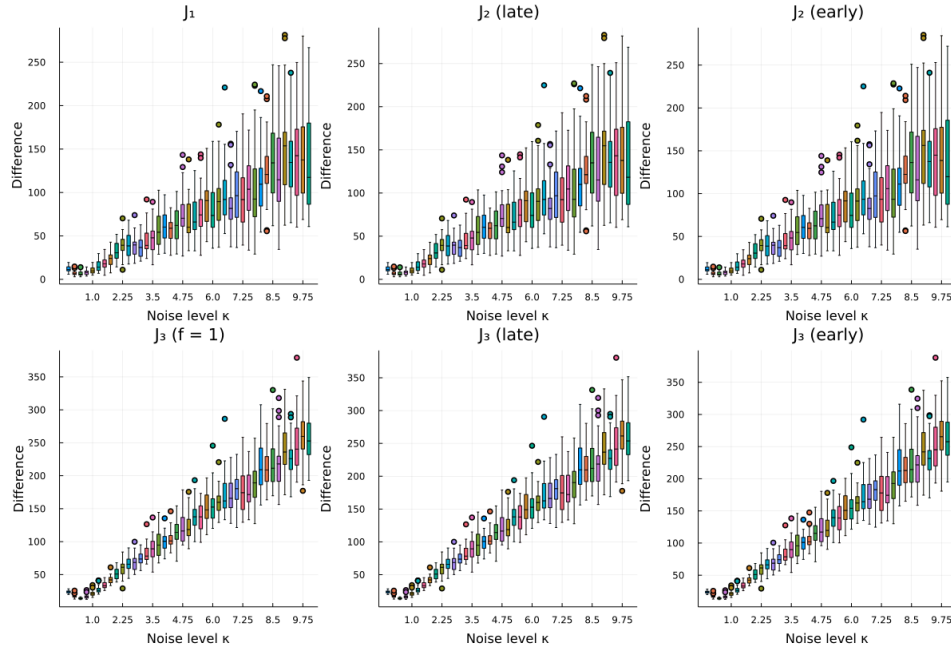

FIGURE 5. Boxplots of the difference between the 1-centrality distance in Figure 4 and corresponding bounds given by Proposition 3.10.

4. Application to Fractal-Like Point Clouds

This chapter explores the application of centrality measures to self-similar point clouds. While toy datasets offer valuable starting points, their applicability to real-world scenarios may be limited. In contrast, fractal-like point clouds, with their inherent complexity and potential for higher dimensionality, have been vastly documented to appear in nature [**fractal**]. Persistent homology has been studied on fractals such as computing the affine fractals from landscapes [**fractalpers**] and estimating the fractal dimension [**fractaldim**]. Analyzing these structures allows us to explore the generalizability and effectiveness of centrality measures in deciphering intricate network-like relationships within spatial data.

For this application, we employ a method for separating signal from noise in persistence diagrams using a hypothesis testing approach developed by Bobrowski and Skraba [**universal**]. This method relies on the concept of *multiplicative persistence* $(\pi(p) = \frac{d}{b})$ for any birth-death pair (b, d) in the kth persistence diagram dgm_k. Throughout this chapter, we operate under the assumption of the validity of the strong universality conjecture. Within this theoretical framework, a left-skewed Gumbel distribution (LGumbel) plays a pivotal role in the hypothesis testing process. In the conjecture below, the constant λ is the Euler-Mascheroni constant. Moreover, $A(\mathcal{T})$ is 1 if $\mathcal{T}$ is the Rips filtration, and 0.5 if $\mathcal{T}$ is the Čech filtration.

Conjecture 4.1. *Consider d-dimensional metric measure space $\mathcal{S}$ and a sequence of random variables $\mathbf{X}_n = (X_1, \ldots, X_n) \in S^n$ with joint probability law $\mathbb{P}_n$. Let $\mathbb{S} = (\mathcal{S}, \mathbb{P})$ be a sampling model under a filtration type $\mathcal{T}$. For any $\mathbb{S} \in \mathcal{U}, \mathcal{T}$, and $k \geq 1$, the limit of $\mathcal{L}_n$ as n approaches infinity equals the left-skewed Gumbel distribution, where*

$$\mathcal{L}_n(\mathbb{S}, \mathcal{T}, k) = \frac{1}{|dgm_k|} \sum_{p \in dgm_k} \delta_{l(p)}$$

and $l(p) = A(\mathcal{T}) \log \log(\pi(p)) - \lambda - \bar{L}$. The expression $\bar{L}$ is given by

$$\frac{1}{dgm_k} \sum_{p \in dgm_k} \log \log(\pi(p)).$$

This conjecture states that, as the sample size approaches infinity, the distribution of certain features derived from persistence diagrams converges to the left skewed Gumbel (LGumbel) distribution.

From the hypothesis testing framework, we recover the signal part dgm_k^S of the diagram by considering only those $p \in \text{dgm}_k$ where

$$e^{-e^{l(p)}} < \frac{\alpha}{|\text{dgm}_k|}$$

for a significance level α. Until the end of this chapter, we consider the significance level $\alpha = 0.05$.

We now explore the topological structure of a point cloud containing 1,000 points sampled around the well-known Sierpinski Triangle (Figure 6). Given the inherent variability in point cloud data and the complex nature of TDA, we employ a bootstrapping approach to assess the robustness of our findings regarding the number of holes (signals) identified.

Bootstrapping [**bootstrap**] involves creating multiple random samples (with replacement) from the original point cloud. In this case, we generated 1,000 samples,
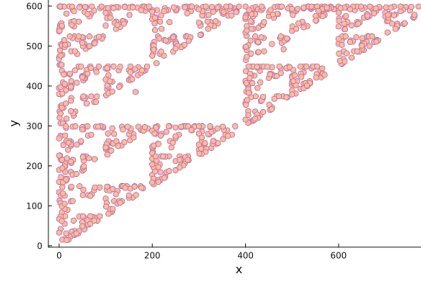
FIGURE 6. A point cloud sampled around a Sierpinski triangle.

each containing 800 points (80% of the original data). This sampling percentage is chosen following the rule of thumb that using at least 50% of the data is beneficial for obtaining statistically significant results in bootstrapping.

We then use the hypothesis testing framework for each bootstrapped sample to identify the number of holes. The mean number of holes across all bootstrapped samples is 0.68, with an approximate standard error of 0.018. The 95% confidence interval for the number of holes is $(0.643, 0.717)$. The confidence interval suggests that the true number of holes in the original point cloud likely falls between 0 and 1. However, the interval is skewed slightly closer to 1, indicating a higher probability of there being a single hole present in the data.

We examine one bootstrapped sample whose plot is shown in Figure 7a. Applying the hypothesis testing framework, we identify only a single hole as statistically significant (signal). The corresponding signal is visualized in Figure 7b. The signal contains points from the largest triangle in the fractal.

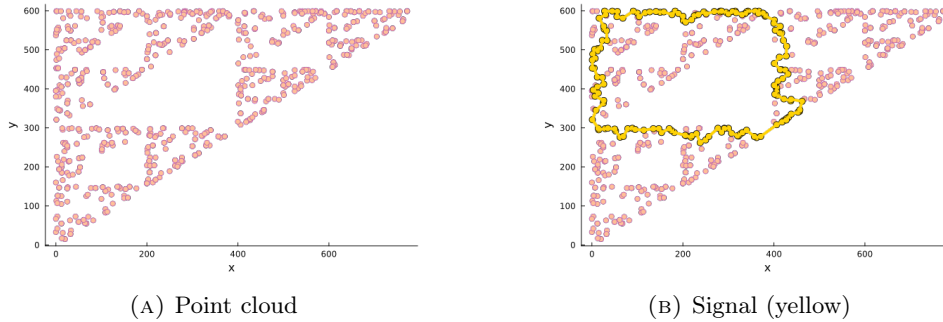(A) Point cloud

(B) Signal (yellow)

FIGURE 7. Signal from a bootstrap sample of the point cloud in Figure 6.

In the context of centrality, we consider two primary ideas for analysis. Direct merging events between holes offer insights into the localized dynamics around the merged region. However, a more comprehensive perspective is gained by considering the cumulative effect of merging across the entire filtration process. This broader analysis, encompassing all merging events, can reveal structures with global importance in shaping the overall landscape of significant holes. Next, while centrality measures rank the holes, a quantitative method to pinpoint the most important ones

might not be readily available. Here, we address this challenge by examining the relative difference in centrality values between the top-ranked holes and the majority of others. Significant differences in centrality would suggest a higher likelihood of those top-ranked holes being truly central features within the data.

Figure 8 depicts the centrality plots associated with the point cloud. To distinguish between our different centrality measures, we denote centrality functions J_3 with various scaling factors as J_4, J_5, and J_6. The function J_4 has factors equal to one, while the functions J_4 and J_5 prioritize late merging and early merging, respectively.
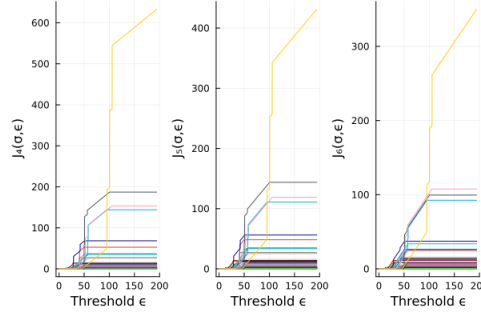
FIGURE 8. Centrality plots of dimension 1 associated to the Rips filtration of the point cloud in Figure 6.

The centrality plots suggest the presence of a relatively important signal within the point cloud data. This is evidenced by the large difference in the maximum centrality value for the highest ranked hole compared to the others. Notably, this hole coincides with the one previously identified using the hypothesis testing framework. This strong agreement between centrality measures and the well-established topological tools found in persistence diagrams reinforces the significance of this particular hole as a key topological feature within the data.

Further bolstering this observation, we can consider persistence values as a ranking system for holes. Computing the Spearman rank correlation coefficient [**spearman**] between the maximum centrality values of J_5 and the persistence values yields a value of 0.997. This value indicates a near-perfect, monotonically increasing relationship between the two rankings. In simpler terms, holes ranked highly by centrality also tend to have high persistence values. By aligning with established methods like persistence diagrams, these findings suggest that centrality functions effectively capture features similar to those identified by common TDA summaries.

We examine other bootstrap sample (Figure 9a) where the hypothesis testing framework fails to identify a signal. Interestingly, the persistence diagram (Figure 9c) exhibits a point far from the diagonal, potentially indicating a feature. However, this point is not classified as a signal by the hypothesis testing framework.

Despite the lack of a signal using traditional methods, both the persistence and the centrality functions, viewed as rankings, reveal some patterns. Mirroring the previous bootstrap sample, the highest ranked hole by both methods corresponds to points from the largest triangle in the fractal. Similarly, the second highest ranked hole aligns with points from the next largest triangle.

To gain a deeper understanding, we delve into the values of both summaries and how they identify holes. We examine how many holes satisfy a threshold, defined

(A) Point cloud

(B) Two highest ranked holes

(C) Persistence diagram of dimension 1
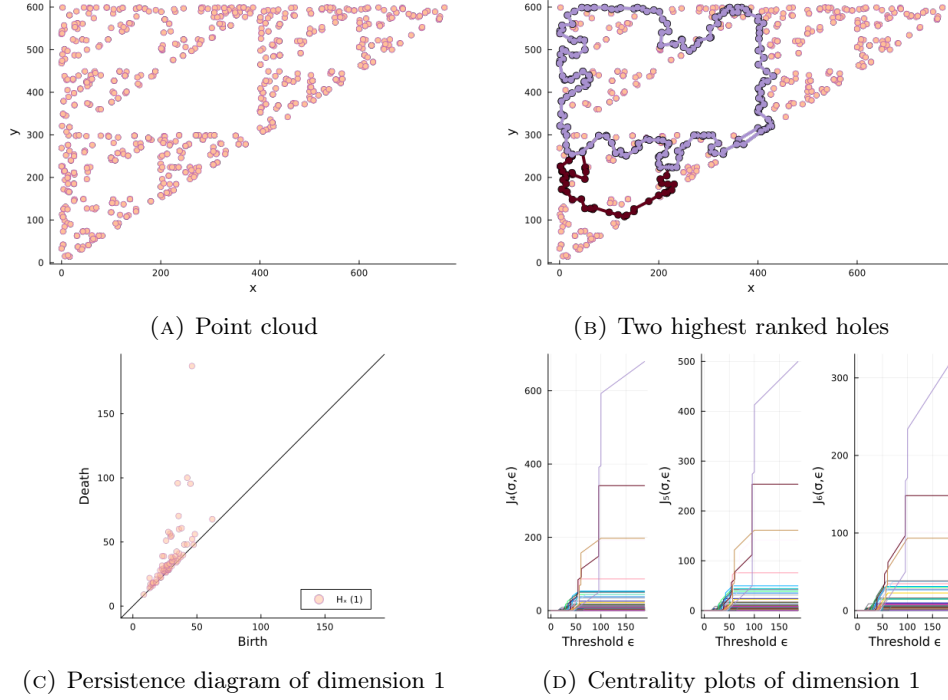

(D) Centrality plots of dimension 1

FIGURE 9. Two highest ranked holes according to the persistence diagram and the centrality plots from a bootstrap sample of the point cloud in Figure 6 with no identified signals.

as a percentage (i) of the maximum value from each summary. We will analyze a range of percentages for i from 0.25 to 1. This analysis will be performed for both the persistence diagram and the centrality function J_5. Figure 10a illustrates the graph of the number of holes identified by each method at different thresholds. The graph reveals interesting differences in how persistence and centrality identify holes. While J_5 identifies two holes meeting the threshold at $i = 0.5$, the persistence diagram identifies a single hole at a slightly lower threshold (around 0.4). As we move to lower thresholds, the persistence diagram consistently identifies more holes than J_5 at each step. These observations suggest that centrality has the tendency to be selective in identifying features compared to the persistence diagram. These observations highlight the potential of centrality functions to capture features that might differ when using common topological summaries.

We can also identify and analyze the merging events across the entire filtration process. Figure 10b shows the earliest homology class considering all possible orders of merging clusters in each of the two highest ranked holes. We note the small diameter of each hole. This observation suggests that some short-lived holes, present during the early stages of filtration, contribute to the overall centrality of the larger holes. Additionally, the centrality of these large holes appears to be influenced by merge instances originating from them much earlier in the filtration process.

We conclude this section by examining a different point cloud, sampled around the Barnsley Fern fractal (Figure 11a). Here, we employ the same bootstrapping
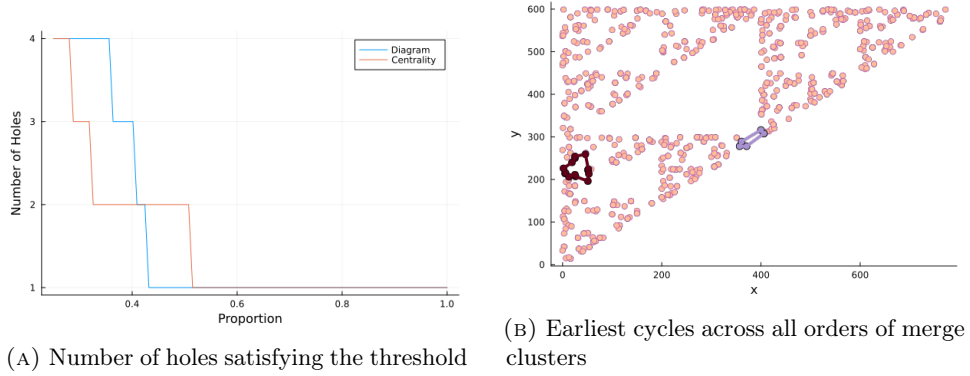
(A) Number of holes satisfying the threshold

(B) Earliest cycles across all orders of merge clusters

FIGURE 10. Additional features that the centrality functions capture from the point cloud in Figure 9a.

(A) Point cloud

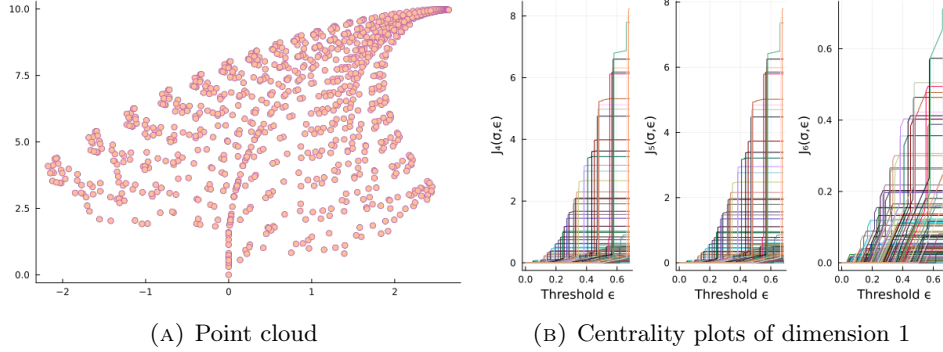(B) Centrality plots of dimension 1

FIGURE 11. Centrality plots, of dimension 1, produced by the Rips filtration of a point cloud sampled around the Barnsley Fern.

approach used previously. No signals were identified in any of the bootstrapped samples for the Barnsley Fern. The Spearman rank correlation between the maximum J_5 centrality values and the persistence values remains high (0.893), indicating a strong monotonically increasing relationship. Similar to the previous example, this suggests that centrality and persistence rankings tend to agree on the relative importance of holes despite the absence of identifiable signals. Furthermore, Figure 12a shows the three highest ranked holes identified by centrality. Extracting the earliest cycles across all possible orders of merge clusters for these three holes reveals a single cycle. In simpler terms, this implies that one hole merged with another, then the survivor merged with the third hole, forming a single connected component at some point during the filtration process. The representative of the single cycle further suggests that the corresponding hole might have played a central role in connecting the other two holes during the filtration process.

Similar to our previous analysis, we compared how centrality and persistence identify holes based on thresholds. Figure 13 illustrates the number of holes detected by each method at different threshold levels (represented as a percentage of the maximum persistence value). The plot reveals key differences in how these methods

(A) Three highest ranked holes

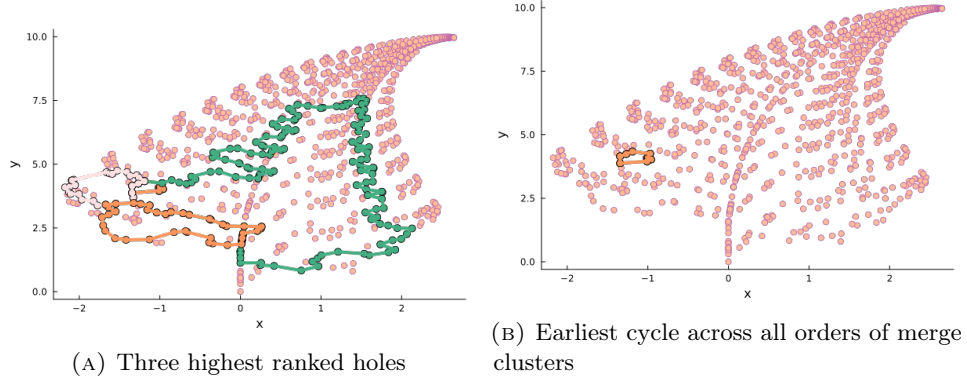(B) Earliest cycle across all orders of merge clusters

FIGURE 12. The earliest cycle across all orders of merge clusters of each of the three highest ranked holes

classify holes. At higher thresholds ($i > 0.6$), the centrality function J_5 identifies holes that meet the threshold. However, the persistence diagram identifies a larger number of holes, including those with lower persistence values ($i \leq 0.6$). This trend continues as we move towards lower thresholds. The persistence diagram consistently detects a higher number of holes (at least 50 holes at $i = 0.25$) compared to the centrality plot (which identifies at most 20 holes at $i = 0.25$). These observations reinforce our earlier finding that the centrality functions might be more selective in identifying features depending on a certain threshold.
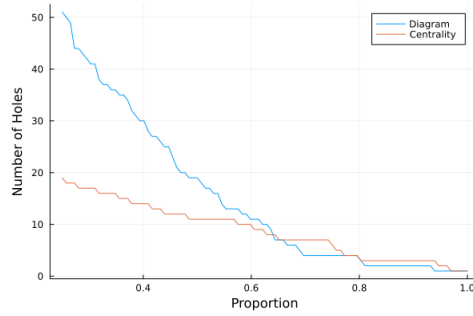

FIGURE 13. Number of holes satisfying the threshold depending on the proportion of the maximum persistence and the maximum J_5 centrality values.

5. CONCLUSION

We introduced novel centrality measures that leverage both persistence and merge dynamics of homology classes. These measures aim to capture a more comprehensive picture of the topological structure within point cloud data compared to traditional summaries. The algorithm for computing the merge dynamics of homology classes is guided by the equivalence of merging and q-nearness between two classes. Similar to persistence barcodes, we have generated shape descriptors in the form of plots and

heat maps and established stability by defining a pseudo-metric similar to bottleneck and landscape distances. Our initial investigation with self-similar point clouds has demonstrated agreement with existing TDA tools, while also revealing additional properties of important features.

Moving forward, we plan to assess the efficacy of these measures across diverse real-world point cloud datasets and within machine learning contexts. Furthermore, future research endeavors will focus on refining the centrality functions and investigating their mathematical properties in greater depth.

Acknowledgements

References

[1] Pierre-Louis Giscard and Richard C. Wilson. "A centrality measure for cycles and subgraphs II". In: *Applied Network Science* 3.1 (June 2018), p. 9. ISSN: 2364-8228. DOI: 10.1007/s41109-018-0064-5. URL: https://doi.org/10.1007/s41109-018-0064-5.

[2] Ernesto Estrada and Grant J. Ross. "Centralities in simplicial complexes. Applications to protein interaction networks". In: *Journal of Theoretical Biology* 438 (2018), pp. 46–60. ISSN: 0022-5193. DOI: https://doi.org/10.1016/j.jtbi.2017.11.003. URL: http://www.sciencedirect.com/science/article/pii/S0022519317305040.

[3] Peter Bubenik et al. "Persistent homology detects curvature". In: *Inverse Problems* 36.2 (Jan. 2020), p. 025008. DOI: 10.1088/1361-6420/ab4ac0. URL: https://doi.org/10.1088/1361-6420/ab4ac0.

[4] Giovanni Petri et al. "Topological Strata of Weighted Complex Networks". In: *PLOS ONE* 8.6 (June 2013), pp. 1–8. DOI: 10.1371/journal.pone.0066506. URL: https://doi.org/10.1371/journal.pone.0066506.

[5] Afra Zomorodian and Gunnar Carlsson. "Computing Persistent Homology". In: *Discrete & Computational Geometry* 33.2 (Feb. 2005), pp. 249–274. ISSN: 1432-0444. DOI: 10.1007/s00454-004-1146-y. URL: https://doi.org/10.1007/s00454-004-1146-y.

[6] Paul Samuel Ignacio, Jay-Anne Bulauan, and David Uminsky. "Lumáwig: An Efficient Algorithm for Dimension Zero Bottleneck Distance Computation in Topological Data Analysis". In: *Algorithms* 13.11 (2020). ISSN: 1999-4893. DOI: 10.3390/a13110291. URL: https://www.mdpi.com/1999-4893/13/11/291.

[7] Peter Bubenik and Paweł Dłotko. "A persistence landscapes toolbox for topological statistics". In: *Journal of Symbolic Computation* 78 (2017). Algorithms and Software for Computational Topology, pp. 91–114. ISSN: 0747-7171. DOI: https://doi.org/10.1016/j.jsc.2016.03.009. URL: http://www.sciencedirect.com/science/article/pii/S0747717116300104.

[8]   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*. SCG '06. Sedona, Arizona, USA: Association for Computing Machinery, 2006, pp. 119–126. ISBN: 1595933409. DOI: `10.1145/1137856.1137877`. URL: `https://doi.org/10.1145/1137856.1137877`.

Department of Mathematics and Computer Science, University of the Philippines Baguio, Baguio City, Philippines 2600
   Email address: `jdmanzanares@up.edu.ph`

Department of Mathematics and Computer Science, University of the Philippines Baguio, Baguio City, Philippines 2600
   Email address: `ppignacio@up.edu.ph`