PLOS Complex Systems

RESEARCH ARTICLE

Persistence partitions of real and synthetic networks

Abigail Jenkins[1], Nick Callor[2], Zachary M. Boyd[1], Taylor Gledhill[1], Raelynn Wonnacott[3], Benjamin Z. Webb[1]*

1 Department of Mathematics, Brigham Young University, Provo, Utah, United States of America,
2 Department of Mathematics, Utah Valley University, Orem, Utah, United States of America,
3 Department of Mathematics, University of Maryland, College Park, Maryland, United States of America

* bwebb@mathematics.byu.edu

Citation: Jenkins A, Callor N, Boyd ZM, Gledhill T, Wonnacott R, Webb BZ (2026) Persistence partitions of real and synthetic networks. PLOS Complex Syst 3(7): e0000109. https://doi.org/10.1371/journal.pcsy.0000109

Editor: Marc Hütt, Constructor University Bremen gGmbH, GERMANY

Received: April 11, 2025

Accepted: May 19, 2026

Published: July 9, 2026

Peer Review History: PLOS recognizes the benefits of transparency in the peer review process; therefore, we enable the publication of all of the content of peer review and author responses alongside final, published articles. The editorial history of this article is available here: https://doi.org/10.1371/journal.pcsy.0000109

Abstract

Determining network structures that are neither local nor global is an area of research that has received considerable attention. The study of these intermediate structures has been primarily concerned with the detection of network communities but also includes the examination of network roles, core and peripheral structure, etc. In an increasingly relevant line of research, persistent homology has also been used to analyze the shape of a network in terms of the network's cycle structure and its higher-dimensional analogues. In this work, we bring these two perspectives together by introducing a non-parametric partition of a network derived from its persistent homology. This partition, which we call the network's persistence partition, is defined using the concept of a *persistence surface*, assigning to each node a measure of its individual persistence relative to its position in the network. We examine the extent to which this partition aligns with standard notions of network roles defined via combinatorial equivalence. We then compare how persistence partitions relate to communities and to the core–periphery structure of a network. Our analysis draws on both real and synthetic networks and demonstrates that persistent homology can reveal distinctive structural features that are not detected by conventional methods.

Author summary

The position a node occupies in a network influences quantities such as the node's importance, role(s), community affiliation(s), etc. within the network. As we show in this paper, this position is possible to characterize relative to the voids and gaps found in the network. Here the voids and gaps refer to places

Data availability statement: The examples in this paper, which include the Gundangborn genealogical dataset, Zachary's karate club, and the New York taxicab network, are available at the given references: [26] Gundangborn Genealogical Dataset. https://www.kinsources.net/kidarep/dataset-81-gundangborn-1948-au02.xhtml Accessed: 2024-10-16. [28] Zachary WW. An Information Flow Model for Conflict and Fission in Small Groups. Journal of Anthropological Research. Vol. 33, Num. 4, 1977: 452–473. [29] New York City Taxi Cab Dataset. https://www.nyc.gov/site/tlc/about/tlc-trip-record-data.page Accessed: 2024-10-31.

Funding: B. W. was partially supported by the National Science Foundation (NSF) https://www.nsf.gov/ grant #2205837 in Applied Mathematics. The sponsors did not play any role in the study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing interests: The authors have declared that no competing interests exist.

left empty by the network's cycles and their higher-dimensional analogues, which are the objects of study of persistent homology. To characterize this homology we introduce the notion of a node's persistence surface, which reflects the order in which each network cycle, etc. are encountered as we move away from the node. Comparing these surfaces naturally leads to a partition of the network: nodes whose persistence surfaces coincide belong to the same class. The resulting persistence-based partition is uniquely determined by the choice of distance function on the network. It provides a way to analyze how a network's persistent homology induces a node partition, and to compare this with partitions obtained from other network-science techniques. This includes node partitions by community, role, core and peripheral structure, etc. Some initial results and discussion relative to real and synthetic networks are described in this paper. A node's position within a network shapes many of its properties, including its importance, functional role, and potential community or core–periphery affiliations. In this paper, we show that this positional information can be characterized in terms of the network's voids and gaps—regions left empty by cycles and their higher-dimensional analogues, which are precisely the structures studied in persistent homology. To capture this relationship, we introduce the notion of a node's persistence surface, which records the order in which the network's cycles and higher-dimensional features are encountered as one expands outward from the node. Comparing these surfaces naturally leads to a partition of the network: nodes whose persistence surfaces coincide (equivalently, nodes that "see" the same homological features in the same order) belong to the same class. The resulting persistence-based partition, which is uniquely determined by the chosen distance function on the network, provides a principled way to understand how persistent homology induces a node equivalence relation. This perspective enables direct comparison with partitions arising from other areas of network science, including those based on communities, roles, core–periphery structure, and more. In this paper, we present initial results and examples illustrating how persistence-based partitions behave on both synthetic and real networks.

1 Introduction

Partitioning the nodes of a network based on notions of similarity is a central theme of network science. These partitions are often used to understand the community, role, and core-periphery structure of networks, among others. In community detection a wide variety of approaches exist to partition the nodes of a network including modularity maximization [1], spectral partitioning [2], and similarity methods [3], as well as other model-based approaches that partition based on creating synthetic communities. This includes stochastic block models [4], random geometric graph models [5], etc. Likewise, role-based methods partition networks into nodes that play similar roles using structural and functional methods including

both deterministic and stochastic approaches [6–13]. Partitioning a network into its core and peripheral structure divides a network into its most central vertices and the surrounding layers. Here, approaches range from using centralities and capacities to modularities, random walks, and reduction methods to determine core-peripheral structures [14–18].

In a very different but increasingly relevant line of research, persistent homology has been used in network analysis to analyze the shape of a network in terms of its "voids" or "gaps" [19]. The idea behind persistent homology involves filling in these gaps while tracking the changes to the network's structure. These gaps often have useful interpretations in the application context, and these methods have become widely used in applications such as the analysis of text systems [20], sensor networks [21], porous material [22], genealogical networks [23], brain networks [24], and others.

In the present work, we combine these two research areas with interesting results. We use local persistent homology features of nodes, given by a node's *persistence surface*, to place nodes into equivalence classes. This notion of a persistence surface is one of the main contributions of this paper. It associates to every node a mapping of the node's individual persistence relative to its position within the network (see Definition 3). If two nodes have equivalent surfaces, they are grouped together creating a nonparametric partition of the network into what we refer to as *persistence partition*.

We examine the extent to which a network's persistence partition aligns with standard notions of network roles, communities, and core–periphery structure. In particular, we compare persistence partitions with measures of combinatorial equivalence—commonly used in social network analysis—to detect role structure, including structural and automorphic equivalence. As a natural extension of automorphic equivalence, we consider two additional role-finding notions, equitable equivalence and latent equivalence, and compare these to persistence equivalence given by the network's persistence partition.

We further investigate how persistence partitions relate to community structure and to core–periphery organization. This analysis draws on examples from genealogical, social, and transportation networks, as well as synthetic benchmarks. Altogether, we provide an extensive comparison of our homological partitioning framework with these well-established partition types, highlighting cases in which our approach is finer, coarser, or simply distinct.

In summary, we make the following contributions:

- Introduce the notion of a persistence surface and its use as a tool for determining persistence partitions of general networks.
- Describe potential applications of persistence partitions in determining the community structure, role structure, and peripheral structure of a network.
- We prove that persistence, equitable, and latent equivalence each yield partitions that are refined by any partition induced by automorphic equivalence.
- Report both similarities and differences between standard community structure and persistence partitions, as well as differences between conventional core–periphery structure and persistence partitions.

The paper is organized as follows. In Section 2, we review the persistent homology of networks, the main tool used to develop the notion of a persistence partition. In this section we introduce the concept of a persistence surface and presents the persistence surface algorithm (PSA), which is used to compute a network's persistence partition. In Section 3, we formalize the persistence partition as an equivalence relation and compare it with both standard and newly introduced methods for determining network roles. In the same section, we examine persistence partitions of several real and synthetic networks, highlighting their similarities and differences with community structure and core–periphery organization. Finally, in Section 4, we summarize our findings and outline several open questions.

2 Materials and methods

The *topology* of a network, which is the structure of the network's connections, is most often represented by a graph. A *graph* $G=(V,E)$ is a collection of *vertices* $V = \{1, 2, \ldots, n\}$ or *nodes*, where vertex i represents the ith element of the network. The *edge set* E consists of the edges $\{i,j\}$ representing an undirected edge between vertices i and j.

More generally, networks can be weighted as well as directed, in the sense that each edge is assigned a weight and/or direction. However, in this paper we only consider unweighted undirected networks, or networks where all interactions are considered to have unit weight and no direction. That said, the work presented in this paper can be applied to weighted, directed, and time-dependent networks in similar fashion. Similarly, although we limit ourselves to shortest paths in this paper, the method we propose can also be applied using any distance metric on a graph. We leave these extensions to others as the point of this paper is to lay the groundwork for describing a method of partitioning a network using persistent homology.

The point of this section is to describe the persistent homology of a network, which will be the main tool we use in this paper to analyze real-world networks. To introduce the notion of persistent homology we require the following setup.

The distance matrix for a network with graph $G = (V,E)$ is given by $D(G) = [d_{ij}]$ where $d_{ij} = d(i,j)$ is the length of the shortest path between vertices i and j. We form a simplicial complex G_δ for each value δ in the distance matrix $D(G)$ as follows. The set of vertices of G is the set of 0-simplices of G_δ. Here, we use the terms *0-simplex* and *vertex* interchangeably and index the 0-simplices the same way as the vertices given that the distinction between the vertices and the 0-simplices is purely formal. As such, we may consider G to be a simplicial complex as well as a graph.

We let the edge $\{i,j\} \in E$ be identified with the 1-simplex formed by i and j. We note that the simplicial complex G_δ may contain objects that do not have equivalent representatives in the network G, namely the n-simplices for $n \geq 2$. For each integer $n \geq 2$, the set of n-simplices in G_δ consists of all n-simplices $[a_0\ a_1 \ldots a_n]$ such that each $a_i \in V$ and $d(a_i, a_j) \leq \delta$ for $0 \leq i < j \leq n$. In other words, G_δ includes an n-simplex σ if each vertex listed in σ is within a distance δ of every vertex in σ.

We extend our definition of G_δ to include all non-negative integers in order to simplify our remaining definitions. Specifically, let δ_i be the greatest entry of $D(G)$ such that $\delta_i \leq i$ for $i \geq 0$ and let $G_i = G_{\delta_i}$. Our construction of G_δ along with this definition ensures the following three important properties hold for all G_i.

1. Once a simplex is included for some G_i, it is present in all later G_j. That is, for $i<j$, $G_i \subset G_j$.
2. The simplicial complex G_1 has exactly the same edges and vertices as G so that $G \subset G_1$.
3. As the graphs we consider are finite, there is a maximal simplicial complex G_M that contains all other G_i. In particular, if $M = \max_{ij} d(i,j)$, then for all i, $G_i \subset G_M$. Furthermore, for $i \geq M$, $G_i = G_M$.

Example 1. ***(Decagonal Theta-Curve Network)*** *Consider the decagonal theta-curve network given by the graph $G=(V,E)$ formed by adding a single edge between the first and fourth vertices of a decagon (see Fig 1(b)). This network has the distance matrix*

$$D(G) = \begin{bmatrix} 0 & 1 & 2 & 1 & 2 & 3 & 4 & 3 & 2 & 1 \\ 1 & 0 & 1 & 2 & 3 & 4 & 5 & 4 & 3 & 2 \\ 2 & 1 & 0 & 1 & 2 & 3 & 4 & 5 & 4 & 3 \\ 1 & 2 & 1 & 0 & 1 & 2 & 3 & 4 & 3 & 2 \\ 2 & 3 & 2 & 1 & 0 & 1 & 2 & 3 & 4 & 3 \\ 3 & 4 & 3 & 2 & 1 & 0 & 1 & 2 & 3 & 4 \\ 4 & 5 & 4 & 3 & 2 & 1 & 0 & 1 & 2 & 3 \\ 3 & 4 & 5 & 4 & 3 & 2 & 1 & 0 & 1 & 2 \\ 2 & 3 & 4 & 3 & 4 & 3 & 2 & 1 & 0 & 1 \\ 1 & 2 & 3 & 2 & 3 & 4 & 3 & 2 & 1 & 0 \end{bmatrix}.$$

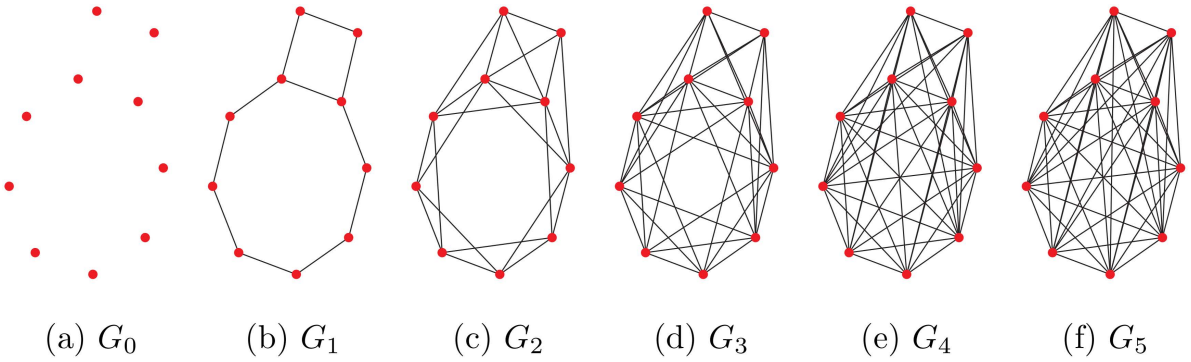

Fig 1. The decagonal theta-curve $G = G_1$ in Example 1 is sequentially filled in by the graphs $G_i = (V, E_i)$ as i increases from 0 to 5. This produces the 1-skeletons of the simplicial complexes G_0, G_1, G_2, G_3, G_4, and G_5; shown left to right.

https://doi.org/10.1371/journal.pcsy.0000109.g001

For the values $i = 0$ through $i = 5 = \max_{ij} d(i,j)$, we form six simplicial complexes G_0, G_1, G_2, G_3, G_4, and G_5 where each G_i includes vertices V, edges E_i, and additional n-simplices as described below. E_0 is empty since no two distinct vertices are at distance 0. Thus G_0 consists of exactly 10 vertices and no higher dimensional simplices. For E_1, we obtain the 11 edges that are present in the original graph. Furthermore, this network has no trivial cycles (i.e., triangles), so G_1 contains no n-simplices for $n \geq 2$. Therefore, $G_1 = G$. Note that there are two minimal-length cycles that generate the first homology group of G, denoted $H_1(G)$; an 8-cycle, η, and a 4-cycle, μ. (For details related to this topic see [25] Chapter 2.)

In G_2 we gain 12 new edges as well as higher-dimensional simplices corresponding to each induced subgraph that is a complete graph. In particular, the induced subgraph from the vertices of μ is complete. Therefore, by construction, μ corresponds to the trivial element of $H_1(G_2)$. In G_3, we gain another 12 new edges and other higher-dimensional simplices. While the vertices of η do not induce a complete subgraph, the G_3 complex is topologically equivalent to a point, and therefore η is equivalent to the trivial element of $H_1(G_3)$. G_4 and G_5 also gain edges and higher-dimensional simplices, but these are also topologically equivalent to a point and therefore introduce no new homology features. Since there are no vertex pairs at distances greater than 5, every subsequent simplicial complex is identical to G_5.

Because each G_i includes the previous complexes, we can study the original structure of G by looking at how the homology changes as i increases. The goal is to understand features of the network that *persist* across multiple values of i. In practice, features that arise from measurement error or random chance typically appear only briefly. The distinction between brief and not brief can be stated exactly according to the Stability Theorem of Persistent Homology (see the Main Theorem of [26]). Roughly speaking, if the error in measuring a network is bounded by some constant $C > 0$, then the persistent homology of the true network and the persistent homology of the noisy network will differ by no more than C. Therefore, any feature that persists for more than C values of i must be a feature of the true network.

Because the G_i simplicial complexes are ordered by inclusion, we also have homomorphisms between each $H_1(G_i)$. If $\sigma \in H_1(G_a)$ is not in the image of $H_1(G_{a-1})$, we say it has a *birth* time a. That is, σ has a representative cycle that first appears in G_a. If $\sigma \in H_1(G_{b-1})$ is nontrivial in $H_1(G_{b-1})$ and is mapped to the trivial element in $H_1(G_b)$, we say it has a *death* time b. That is, every representative cycle for σ is topologically equivalent to the trivial cycle in each G_i with $i \geq b$. In this case, we say that σ has a *persistence interval* $[a,b]$. If σ has birth time a, but is never mapped to a trivial element in $G_{i>a}$, the death time is infinite and the persistence interval is $[a, \infty)$. For our purposes, it will suffice to consider the multiset of persistence intervals for n-cycles as the nth persistent homology group of G, denoted $PH_n(G)$. We note that the persistent homology of G has much more structure, but it is not necessary for our construction in this paper.

In Example 1 the persistence intervals are given by $PH_0(G) = \{[0,1] \times 9, [0,\infty) \times 1\}$, corresponding to the 10 vertices in no particular order; $PH_1(G) = \{[1,2] \times 1, [1,3] \times 1\}$, corresponding to the 4-cycle, μ, and the 8-cycle, η, respectively; and for $n \geq 2$, $PH_n(G) = \emptyset$.

2.1 Equivalence of persistence surfaces

The notion of a network's persistence curve was recently introduced to study the differences among complex networks [23]. Here, we build on this concept, introducing the idea of a persistence surface (see Definition 3). The goal is to use these surfaces to determine a nonparametric partition of the network nodes that respects the network's homotopic structure.

2.2 Persistence curves

Definition 1. ***(Persistence Curves)*** *Let G=(V,E) be the graph of a network with nonempty vertex and edge sets. For each $n \in \mathbb{N}$, consider the set of all persistence intervals, $\{[a_j, b_j] | \sigma_j \in PH_n(G)\}$. Note that when $b_j = \infty$, the interval is half-open: $[a_j, b_j = \infty)$. Without loss of generality, we order σ_j so that for $b_j < \infty$, $b_{j-1} - a_{j-1} \leq b_j - a_j$ and if $b_k = \infty$ then $j < k$. The dimension-D persistence curve of $PH_n(G)$ is the linear interpolation in $\mathbb{R}^2$ of the set of points $\{(b_j - a_j + n, j)\}$.*

Persistence curves allow us to visualize how many intervals of a given persistence there are in the network. This is demonstrated in the following example.

Example 2. ***(Persistence Curves of the Gundangborn Genealogical Network)*** *The Gundangborn network shown in Fig 2 (left) is a genealogical network consisting of thirty-two individuals where edges indicate either a parent-child or union relationship [27] (see S1 Text and S1 Data). The dimension-1 persistence intervals of the network are $\{[1,2] \times 1, [1,3] \times 1\}$. The corresponding cycles are shown in Fig 2 (left) in red and blue, respectively. The intervals are both shown in blue in Fig 2 (right), with the corresponding birth and death persistence curves in purple and cyan, respectively. The death persistence curve is constructed by outlining (interpolating) the upper endpoints of the persistence intervals, and the birth persistence curve outlines the lower endpoints.*

2.3 Persistence surfaces

A network's persistence curve gives us a global picture of its structure, describing how the network's cycles are related (or their equivalent higher-dimensional analogs). Starting from a given vertex $v \in V$ it is possible to use this notion to describe the homological structure of the network as one moves farther from v. The result is the local persistence structure of the network as viewed from the point of view of a specific vertex. To define this structure, we will use the concept of an ego-graph, which allows us to view the network centered at a specific vertex restricted to a specific radius.

Definition 2. ***(Ego-Graph)*** *Let G=(V,E) be a network, and let $v \in V$. The ego-graph $E_v(r) = (\mathcal{V}_r, \mathcal{E}_r)$ of G centered at v is the induced subgraph of G with vertex set $\mathcal{V}_r = \{i \in V : d(v,i) \leq r\}$, i.e., all vertices reachable by a path of length at most r from vertex v, and all edges that exist between these vertices in the original network G.*

In general, for a network $G=(V,E)$ and a vertex $v \in V$ we can always form the sequence of persistence curves for the ego-graphs $E_v(0), E_v(1), \ldots, E_v(d) = G$ where $d < \infty$ is the diameter of the network. If $d = \infty$, i.e., G is disconnected, we can simply restrict our attention to the graph's connected components where the diameter is finite. This sequence of networks can be used to create a persistence surface of the vertex v in the following way.

Definition 3. ***(Persistence Surface)*** *Let G=(V,E) be a network of diameter $d < \infty$ and let $v \in V$. The persistence surface $S_v(G)$ of v is the linear interpolation of all persistence curves of the ego-graphs $E_v(i)$ for $i = 0, 1, \ldots, d$ in this order.*

Example 3. ***(Ego-Graph of the Gundangborn Network)*** *Consider the Gundangborn network G=(V,E) in Fig 2 (left) with the vertex $v \in V$. In Fig 3 the ego-graphs $E_v(0), E_v(1), \ldots, E_v(7) = G$ are shown from top to bottom on the left-hand*

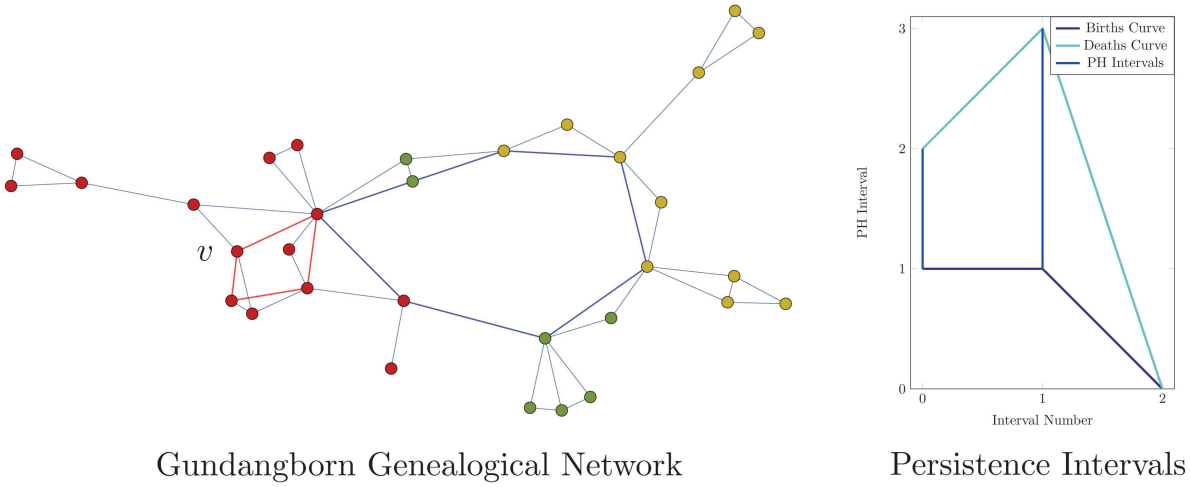

Fig 2. Left: Genealogical network for the Gundangborn of Arnhem Land, Australia consisting of thirty-two individuals (see). The relations are of two types: either parent-child relations or union relationships but are not distinguished here. The cycles generating the [1,2] and [1,3] persistence intervals are shown in red and blue, respectively. Right: The dimension-1 persistence intervals $\{[1,2]\times 1, [1,3]\times 1\}$ and the birth and death persistence curves for the network are shown in blue, purple, and cyan, respectively.

https://doi.org/10.1371/journal.pcsy.0000109.g002

side of the left column, where the vertex v is labeled. On the right side of the column are the persistence curves corresponding to each ego-graph, respectively, from top to bottom.

To create the persistence surface $S_v(G)$ we arrange the persistence curves along the y-axis, which indicates the ego-graph radius. The other axes indicate the persistence interval index (x-axis) and the birth or death time (z-axis). This results in a piecewise linear surface. Note that for a fixed radius $r \in \{0, ..., 7\}$, the cross-section $y = r$ produces the persistence curve of the ego-graph $E_v(r)$ at radius r.

In an effort to improve the appearance of the surface, we choose a viewing window of $[0, I] \times [0, R + 1] \times [0, T + 1]$ where I is the number of persistence intervals in $PH_n(G)$, R is the maximum distance from v to any other vertex, and T is the maximum finite death time of any persistence interval in $PH_n(G)$. The process of first creating the ego graphs then persistence curves, and then the persistence surface is indicated in Fig 3. The result is the persistence surface $S_v(S)$ shown at the bottom right. Note that the purple and cyan colors represent the persistence curves and surfaces created when using the birth and death times of the persistence intervals, respectively.

To create a vertex partition of the graph $G=(V,E)$ we require the following characterization of a persistence surface in terms of what we refer to as *ledges*.

Definition 4. ***(Persistence Surface Ledge)*** *Let $\{E_v(r)\}_{r=0}^{d}$ be the set of ego-graphs for a vertex $v \in V$ in a network given by G = (V,E), where $d < \infty$ is the diameter of G. Let $i, i + 1 \in \{0, ..., d\}$ be such that the multiset of persistence intervals present in $E_v(i)$ is not equivalent to the multiset of persistence intervals present in $E_v(i + 1)$. In other words, the ego-graph $E_v(i + 1)$ contains at least one additional persistence interval more than the ego-graph $E_v(i)$. If this is the case a ledge is formed at radius $i+1$. This ledge is visually represented by a ridge or shelf in the persistence surface at $r = i+1$.*

The sequence of ledges is the list of radii $r_1, \ldots, r_\omega$ at which ledges form as r increases from 0 to d. The sequence of ledge heights $h_1, \ldots, h_\omega$ is the death times h_i of the longest persistence interval corresponding to the ledge r_i for $i = 1, \ldots, \omega$.

In dimensions one and two, a vertex's persistence surface measures a vertex's relation to the cycles and closed singular surfaces in the network, respectively, as we move further and further from the vertex. This allows us to incrementally observe the persistent homology of the network as we increase our distance from a given vertex. Specifically, each ledge

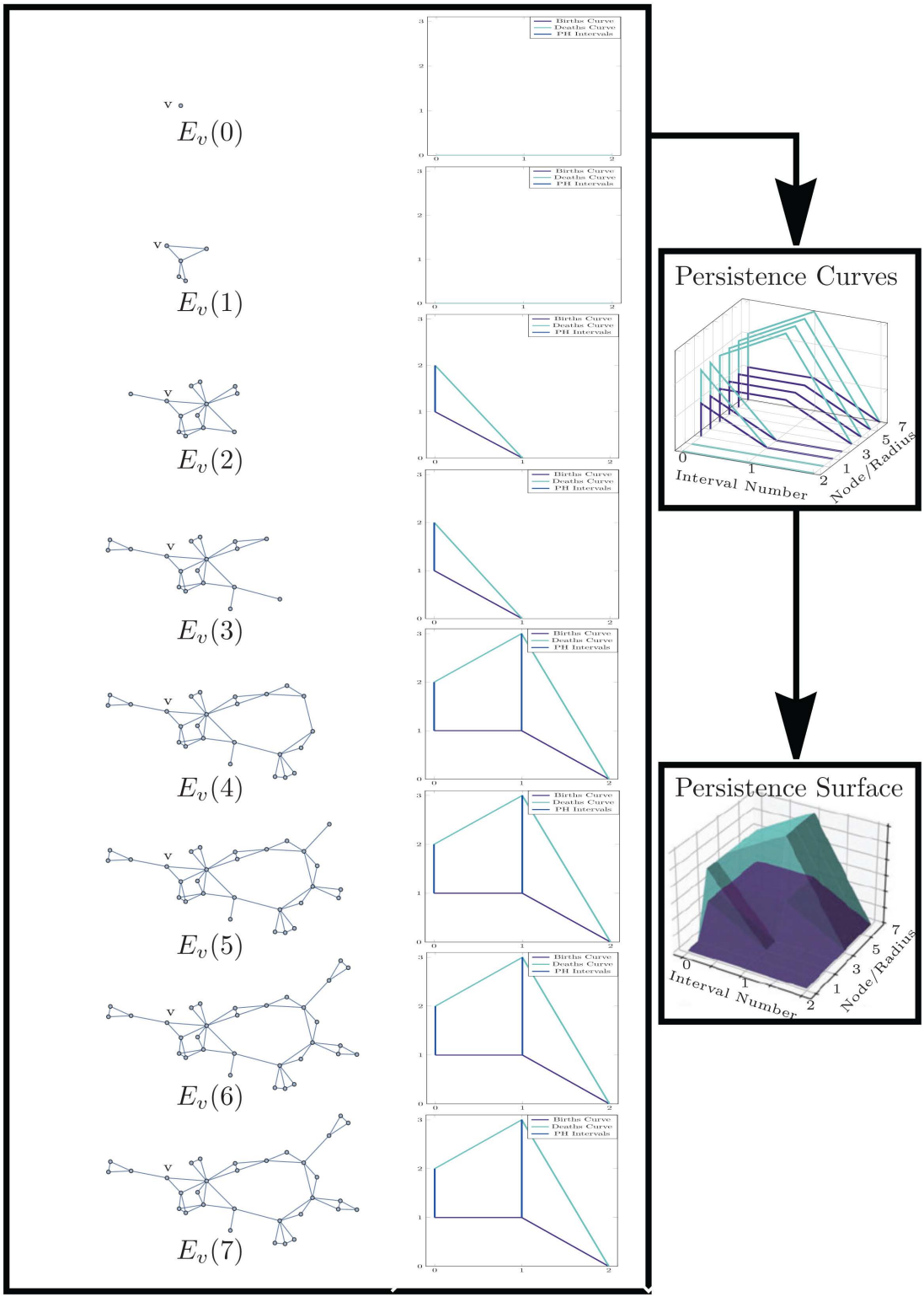

Fig 3. Left: The ego-graphs $E_v(r)$ for the vertex v in the Gundangborn network (see Fig 2, top) are shown left for $r = 0, \ldots, 7$ together with the corresponding persistence curve, shown right. Following the arrows, the persistence surface $S_v(G)$ is constructed by stacking the persistence curves (see Example 3 for details). The purple and cyan surfaces are the persistence surface created using birth and death times, respectively.

https://doi.org/10.1371/journal.pcsy.0000109.g003

at radius r_i of the surface corresponds to a different cycle or set of cycles with death-time h_i. Since the surface is ordered in such a way that the radius increases at each step, the ledges occur in order (moving along the x-axis) of the cycles that become trivial as the radius of the ego-graph increases.

We note that just as persistence intervals belong to a specific dimension, so do persistence curves and persistence surfaces. To simplify illustrating our method, we consider only persistence surfaces for dimensions one and two in this paper. The dimension one persistence surfaces of the Gundangborn network are considered in the following example.

Example 4. ***(Persistence Surfaces of the Gundangborn Genealogical Network)*** *The Gundangborn network $G=(V,E)$ has $|V|=34$ individuals each with their own persistence surface $S_i(G), i = 1, ..., 34$. The surfaces of the vertices can be grouped according to type (see Fig 4):*

i. *the surface has two ledges that both remain in line with the top of the surface;*

ii. *the surface has one ledge; and*

iii. *the surface has two ledges, the first of which peaks above the top of the surface before dropping back down.*

These surfaces suggest that each individual in the network plays one of three roles relative to the network's persistent homology within the network corresponding to (i)-(iii). If we color these individuals in (i)-(iii) red, green, and yellow, respectively, the result of this partition is shown in Fig 2 (top). The red vertices see the five-cycle "first," represented by the first

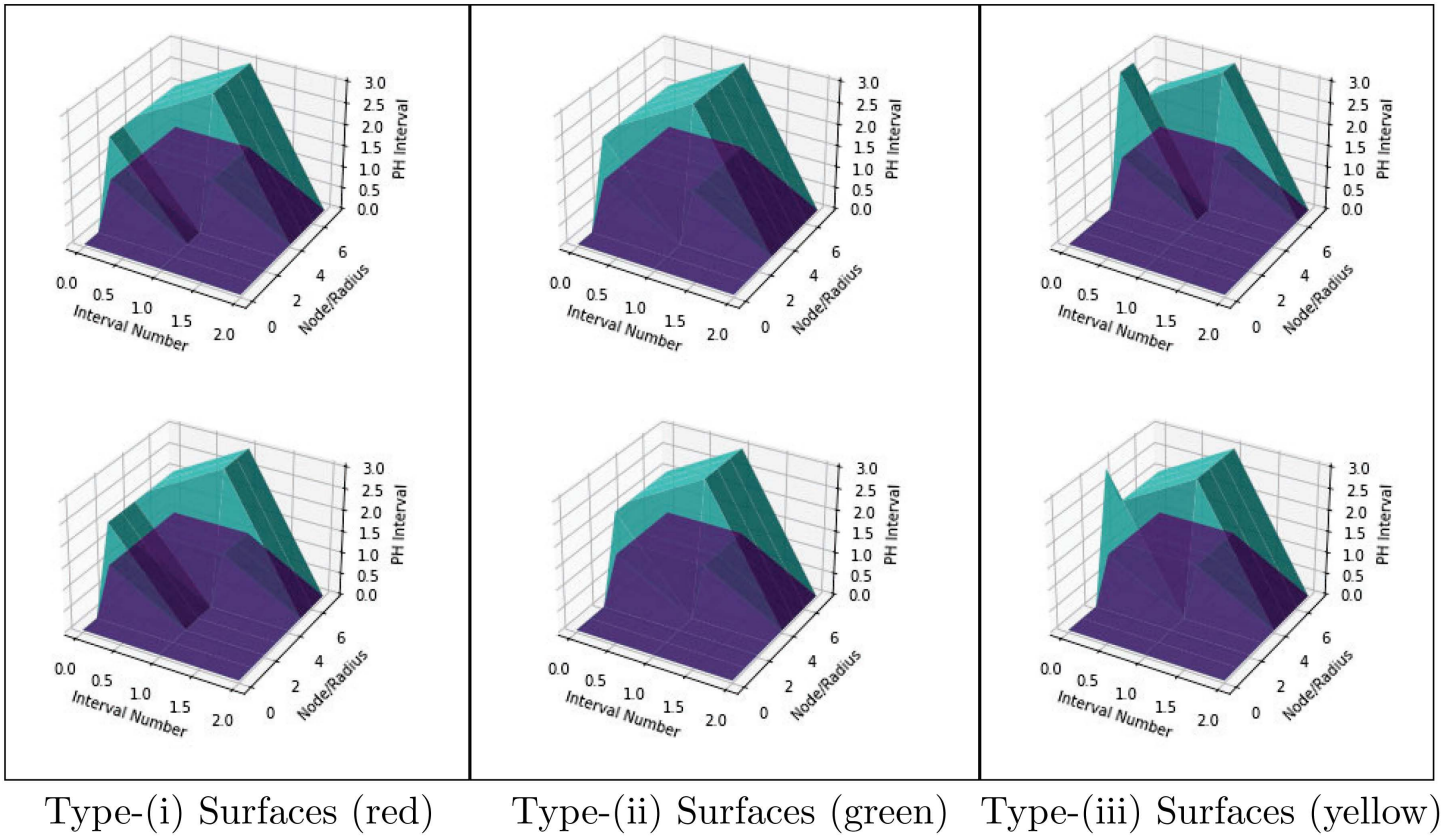

Fig 4. Shown left, center, and right are two distinct persistent surfaces for each of the surface types (i)-(iii) of the Gundangborn network considered in Example 2. Each surface type corresponds to two different vertices in Fig 2 (top) colored red, green, and yellow; respectively.

https://doi.org/10.1371/journal.pcsy.0000109.g004

ledge, and then "later" see the larger seven-cycle represented by the second ledge. Each of the green vertices see both cycles at the same time-step, corresponding to the single ledge of their surfaces. The yellow vertices see the seven-cycle first, corresponding to the first large ledge, and then later see the smaller five-cycle corresponding to the second ledge. Thus, the types of persistence surfaces reflect the order in which each vertex first "sees" the cycles in a given network in terms of distance.

The vertex partition described in Example 4 is relative to the cycle structure or higher-dimensional analogues of the network. In Section 3, we consider how well the elements of this partition align with the standard notions of communities and roles in real and synthetic networks. In the following, we formalize the method used in Example 4 to partition the vertex set of the Gundangborn network.

2.4 Persistence surface partitioning algorithm

To formalize the grouping of surfaces described in Example 4 we define an equivalence relation on the set of persistence surfaces of a network.

Definition 5. ***(Persistence Partition)*** *Suppose $G=(V,E)$ has vertices $i,j \in V$. The persistence surfaces $S_i(G)$ and $S_j(G)$ are topologically equivalent if they have the same sequence of ledge heights $h_1, \ldots, h_\omega$. In this case, we write $S_i(G) \sim S_j(G)$. This is an equivalence relation that induces a partition $\{P_1, \ldots, P_k\}$ of V, where each P_i is a collection of vertices with topologically equivalent surfaces. For fixed dimension $D \geq 1$ we refer to this collection $(P_i)_{i=1}^k$ as the persistence partition of G, where each $P_i \subset V$ is a persistence cell of G.*

We refer to the process of finding a network's persistence birth surfaces and using these to determine the network's persistence partition as the Persistence Surface Algorithm (PSA). We use the program *Ripser* (from the Python package Ripser) [28] to calculate the persistence intervals of a network given by $G=(V,E)$. Ripser has a computational and spatial complexity of $O((n+m)^3)$ where $n=|V|$ and $m=|E|$. Here, the number $n+m$ is the number of network simplices.

Using the Persistence Surface Algorithm requires that we calculate the persistent homology for each ego-graph in the network, resulting in the Ripser algorithm being run at most d times, where d is the diameter of the network. Because this is repeated for each node, finding the network's persistence surfaces has a computational and spatial complexity $O(dn(n+m)^3)$. While this may be prohibitively expensive for large networks, we note that a network's persistence partition can be approximated by limiting the size of a vertex's ego-graph to some maximal radius $r_M < d$. This yields what may be viewed as a *local persistence partition*, whose computation has temporal and spatial complexity $O(r_M n_M (n_M + m_M)^3)$ where n_M denotes the maximum number of vertices in an ego–graph $E_v(r_M)$ over all $v \in V$, and m_M denotes the maximum number of edges among these ego-graphs.

A natural question is whether persistence partitions capture traditional community structure or, alternatively, particular role structures or core–periphery organization within a network. This is investigated in the following section.

3 Results

In the preceding section, we introduced the notion of a persistence partition and presented the persistence surface algorithm (PSA) as a means of computing this partition. In the present section, we investigate the relationship between persistence partitions and both classical and newly developed approaches to role detection—a methodological framework frequently employed to identify structural roles in social networks (see, for instance, [9,10,13]). To facilitate this comparison, we formalize the concept of persistence equivalence, whereby vertices contained in the same persistence cell are deemed equivalent, i.e., they occupy an identical role with respect to the network's persistence structure (see Definition 8). We then compare persistence equivalence with standard equivalence notions, as well as with the new equivalence relation introduced here—latent equivalence—which extend the classical notion of automorphic equivalence (see Section 3.1). Finally, we analyze persistence partitions produced by PSA for several network models known to exhibit nontrivial community structure, and we conclude with a detailed examination of persistence partitions for two empirical networks:

the Zachary Karate Club and the New York City taxi network. For the Zachary Karate Club we also compare its persistent partition to its core-periphery structure (see Section 3.2).

3.1 Persistence equivalence and role detection

Classical approaches to role detection in social networks are grounded in positional analysis, which seeks to group vertices (actors) that share similar patterns of relationships with other vertices (actors). These methods partition network vertices into roles based on various equivalence relations [9,10,13].

More recently, new techniques have been developed that assign roles to vertices using stochastic equivalence and network embeddings. Stochastic equivalence partitions vertices according to a probability distribution: two vertices share the same role if exchanging them yields a graph with the same distribution. Network embedding methods, in contrast, learn low-dimensional representations of networks to uncover role structures, employing tools such as low-rank matrix factorization, random-walk–based algorithms, and deep learning methods [11].

Among these approaches, the methods most closely aligned with the concept of a persistence partition are those that define explicit equivalence relations. In each, the vertices of a network $G=(V,E)$ are partitioned into the sets $V = (V_i)_{i=1}^k$ where the vertices in V_i are assumed to play the same role for each $i = 1, \ldots, k$. The principal forms of this type of equivalence are structural equivalence, automorphic equivalence, and regular equivalence, with structural equivalence being the most restrictive (see [9], for instance).

Definition 6. ***(Structural Equivalence)*** *For the graph $G = (V,E)$ let $\Gamma(v)$ be the set of neighbors of $v \in V$. Two vertices $u, v \in V$ are structurally equivalent if $\Gamma(u) = \Gamma(v)$.*

Since structural equivalence is an equivalence relation, the vertex set V of the graph $G=(V,E)$ is partitioned into equivalence classes $S_1, \ldots, S_k$. We refer to this collection $(S_i)_{i=1}^k$ as the *structural partition* of G, where each $S_i \subset V$ is a *structural* cell of G.

The second type of equivalence considered in the literature is automorphic equivalence which is related to graph symmetries, which are known to be ubiquitous in real-world networks [29]. Symmetries of a graph $G=(V,E)$ can be understood in terms of automorphisms $\phi : V \to V$ that preserve adjacencies. More intuitively, an automorphism describes how parts of a graph can be interchanged without affecting the structure of the graph. These parts (i.e., subgraphs) are symmetric and together with the automorphism ϕ constitute a graph symmetry or vertices that are equivalent with respect to symmetry.

Definition 7. ***(Automorphism Equivalence)*** *An automorphism $\phi : V \to V$ of a graph $G = (V,E)$ is a permutation of V such that the adjacency matrix $A=A(G)$ satisfies*

$$A_{ij} = A_{\phi(i)\phi(j)}$$

for each pair of vertices i and j. The orbit of a vertex $v \in V$ is the vertex set

$$O(v) = \left\{\phi^i(v) : i = 1, 2, ..., \ell \text{ for } \ell = |\phi|\right\}$$

where ℓ is the smallest positive integer such that ϕ^ℓ is the identity. Vertices in the same orbit are automorphically equivalent with respect to ϕ.

If $G=(V,E)$ has an automorphism ϕ, the orbits of ϕ partition the vertices of G into the equivalence class of automorphic vertices $A_1, \ldots, A_k$. This is referred to as the *automorphic partition* $(A_i)_{i=1}^k$ *induced* by ϕ, where each A_i is an *automorphic cell* of G. As a graph may have many graph automorphisms it may have many different automorphic partitions. Our interest in both structural and automorphic equivalence is whether such vertices are persistence equivalent.

Definition 8. ***(Persistence Equivalence)*** *For the graph $G = (V,E)$ let $(P_i)_{i=1}^k$ be its persistence partition. Vertices in the same persistence cell P_i are said to be persistence equivalent for $i = 1, 2, \ldots k$.*

A partition P is a *refinement* of the partition Q if every cell (element) of P is a subset of some cell (element) of Q. Two partitions are *incomparable* if neither is a refinement of the other. It is known that structural equivalence is *stricter* than automorphic equivalence meaning that if $(A_i)_{i=1}^k$ is an automorphic partition of G then it is a refinement of its structural partition $(S_{j=1}^{\ell})$ [13]. Similarly, automorphic equivalence is stricter than persistence equivalence.

Theorem 1. ***(Persistence and Automorphic Equivalence)*** *Suppose $(A_i)_{i=1}^k$ is an automorphic partition of the graph G. Then $(A_i)_{i=1}^k$ is a refinement of the persistence partition $(P_i)_{i=1}^{\ell}$ of G in any dimension $D \geq 1$.*

Proof. Let the graph $\bar{G} = (V, \bar{E})$ where the edge set $\bar{E} = \{\{\phi(i), \phi(j)\} | \{i, j\} \in E\}$. Since $A_{ij} = A_{\phi(i)\phi(j)}$ then $G \cong \bar{G}$ or G is isomorphic to $\bar{G}$ meaning G is the graph $\bar{G}$ except the vertices of $\bar{G}$ have been relabeled. Thus, if $\phi(\alpha) = \beta$ then both $\alpha, \beta \in V$ have isomorphic ego-graphs $E_\alpha(r) \cong E_\beta(r)$ for all $r \geq 0$. Because the ego-graphs determine the persistence curves in any dimension $D \geq 1$ which in turn determine the persistence surface, then both $\alpha \in V$ and $\beta \in V$ have the same persistence surface and are in the same partition element of the network's persistence partition.

Given that the relation $\alpha \sim \beta$ if $\phi(\alpha) = \beta$ is an equivalence relation on the vertices of V, then all vertices in the same orbit of ϕ are in the same persistence cell. □

As automorphisms preserve adjacencies between vertices of a graph, Theorem 1 indicates that if two vertices are symmetric, then they are structurally indistinguishable via some automorphism. Therefore, each ego-graph of each vertex is structurally identical, so both vertices have the same persistence surface and are therefore in the same persistence cell.

With this result in mind, we consider two generalizations of symmetries: equitable partitions and latent symmetries. Equitable partitions have a history of being used in network analysis to partition networks into distinct roles (see, for instance, [30]). Latent symmetries, however, represent a new class of symmetries that have been recently developed [31] but have yet to be applied to role analysis.

While classically associated with the spectral properties of a graph [32], equitable partitions have more recently been connected with dynamical phenomena such as synchronization and network growth [33–36]. The definition is as follows:

Definition 9 (*Equitable Partition*). *An equitable partition of a graph G=(V,E) with adjacency matrix $A \in \{0,1\}^{n \times n}$ is a partition $(E_i)_{i=1}^k$ of V with the property that for all $a, b \in \{1, 2, \ldots, k\}$, the sum*

$$\sum_{j \in E_b} A_{ij} = s_{ab}$$

is constant for any $i \in E_a$.

For a simple graph $G=(V,E)$, an equitable partition can be thought of as dividing the vertices of the graph into the partition $(E_i)_{i=1}^k$ where every vertex in E_a has the same number of neighbors in E_b for all $a, b \in \{1, 2, \ldots, k\}$. A graph can have many different equitable partitions. To define the notion of equitable equivalence we use the a graph's coarsest equitable partition, which has the useful property that it is unique for any graph [37].

Definition 10. ***(Equitable Equivalence)*** *An equitable partition $(E_i)_{i=1}^k$ of a graph G is the graph's coarsest equitable partition if, for any other equitable partition $(E_j^*)_{j=1}^{\ell}$ on G, the cell $E_j^* \subseteq E_i$ for some $j \in \{1, \ldots, k\}$. Vertices in the same cell E_i of the coarsest equitable partition are said to be equitably equivalent.*

We refer to the elements of a graph's coarsest equitable partition as its *equitable cells*. As shown in [32], every automorphism $\phi : V \to V$ induces an equitable partition. This implies that automorphic equivalence is stricter than equitable equivalence.

Theorem 2. ***(Equitable and Automorphic Equivalence)*** *Suppose $(A_i)_{i=1}^k$ is an automorphic partition of the graph G. Then $(A_i)_{i=1}^k$ is a refinement of the coarsest equitable partition $(E_i)_{i=1}^{\ell}$ of G.*

Proof. The automorphic partition $(A_i)_{i=1}^k$ of G has the property that the set $A_1, \ldots, A_k$ is an equitable partition of G. Since any equitable partition is a refinement of the graph's coarsest equitable partition $(E_i)_{i=1}^{\ell}$ by Definition 10 the result follows. □

A second generalization of the notion of a graph symmetry is the notion of a latent symmetry. Latent symmetries are defined as automorphisms of a network with graph $G=(V,E)$ after it has been transformed via an isospectral reduction. Isospectral reductions were first developed in [38] and have been used more recently to improve eigenvalue estimates, determine the dynamic stability of dynamical systems, perform link prediction on real-world networks, characterize cospectral vertices, etc. (see [39–42]). An isospectral reduction of a graph (matrix) is a transformation that reduces the size of a graph while preserving its *spectral properties*, e.g., eigenvalues and eigenvectors (see [43] for a detailed description and further applications).

An isospectral reduction of a graph (matrix) is defined as follows:

Definition 11. ***(Isospectral Reduction)*** *The isospectral reduction of a matrix $M \in \mathbb{R}^{n\times n}$ over the proper subset $S \subseteq N = \{1, 2, \ldots, n\}$ is the matrix*

$$R_S(M) = M_{SS} - M_{S\bar{S}}(M_{\bar{S}\bar{S}} - \lambda I)^{-1} M_{\bar{S}S} \in \mathbb{W}^{|S|\times|S|},$$

where $\bar{S}$ is the complement of S in N. The isospectral reduction of a graph G=(V,E) with adjacency matrix A over the set $S \subset V$ is defined as the graph $R_S(G)$ with adjacency matrix $R_S(A)$.

In Definition 11 the set $\mathbb{W}$ is the set of rational functions in the variable λ. The parameter λ is used because the eigenvalues of the matrix $M \in \mathbb{R}^{n\times n}$ and the smaller reduced matrix $R_S(M) \in \mathbb{W}^{|S|\times|S|}$ are typically the same. In order not to violate the Fundamental Theorem of Algebra, the entries of $R_S(G)$ cannot be scalars but instead are objects that can carry more information.

Definition 12. ***(Latent Equivalence)*** *Suppose $R_S(G) = (\mathcal{V}, \mathcal{E})$ is an isospectral reduction of the graph G=(V,E). If there exists an automorphism $\phi_S : \mathcal{V} \to \mathcal{V}$, then the vertices of $\mathcal{V}$ that are in the same orbit of ϕ_S are referred to as latently equivalent.*

The notion of latent equivalence is based on the idea of a latent symmetry described in [31]. It can be shown that latent equivalence defines an equivalence relation. Moreover, consistent with previous results, automorphic equivalence is stricter than latent equivalence, at least for the undirected graphs considered here.

Theorem 3. ***(Latent and Automorphic Equivalence)*** *Latent equivalence defines an equivalence relation on the vertices of a graph. Furthermore, given an automorphic partition $(A_i)_{i=1}^{k}$ of the graph G, this partition is a refinement of the latent partition $(L_i)_{i=1}^{\ell}$ of G.*

Proof. Corollary 3.4 of [42] states that two vertices in an undirected graph are latently symmetric if and only if they are cospectral. As cospectrality is an equivalence relation the result follows. □

Example 5. ***(Automorphic, Persistent, Equitable, and Latent Partitions)*** *Consider the graph G=(V,E) in Fig 5 shown in each of the four panels. The graph has the automorphic partition $(A_i)_{i=1}^{6}$, the persistence partition $(P_i)_{i=1}^{3}$, the coarsest equitable partition $(E_i)_{i=1}^{1}$, and the latent partition $(L_i)_{i=1}^{5}$. Vertices in the same cell of a given partition share the same color. The automophic partition, shown top center, is a refinement of each of the other partitions as guaranteed by Theorems 1–3. As the graph G is a regular graph its coarsest equitable partition is a single set $E_1 = V$.*

In Fig 5, the latent partition of the graph constitutes a refinement of its persistence partition, which itself refines the coarsest equitable partition of the graph. This ordering need not hold universally. As illustrated in Fig 6, the persistence and latent partitions may be incomparable. It remains an open problem to determine whether any nontrivial structural relations exist among a graph's persistence, coarsest equitable, and latent partitions, such as those established for automorphic and related partitions in Theorems 1–3.

Since structural equivalence is stricter than automorphic equivalence combining this with Theorems 1–3 yields the following corollary.

Corollary 1. ***(Role Equivalence and Refinements)*** *For a graph G=(V,E) the automorphic, persistence, equitable, and latent partitions of G are all refinements of the structural partition of G.*

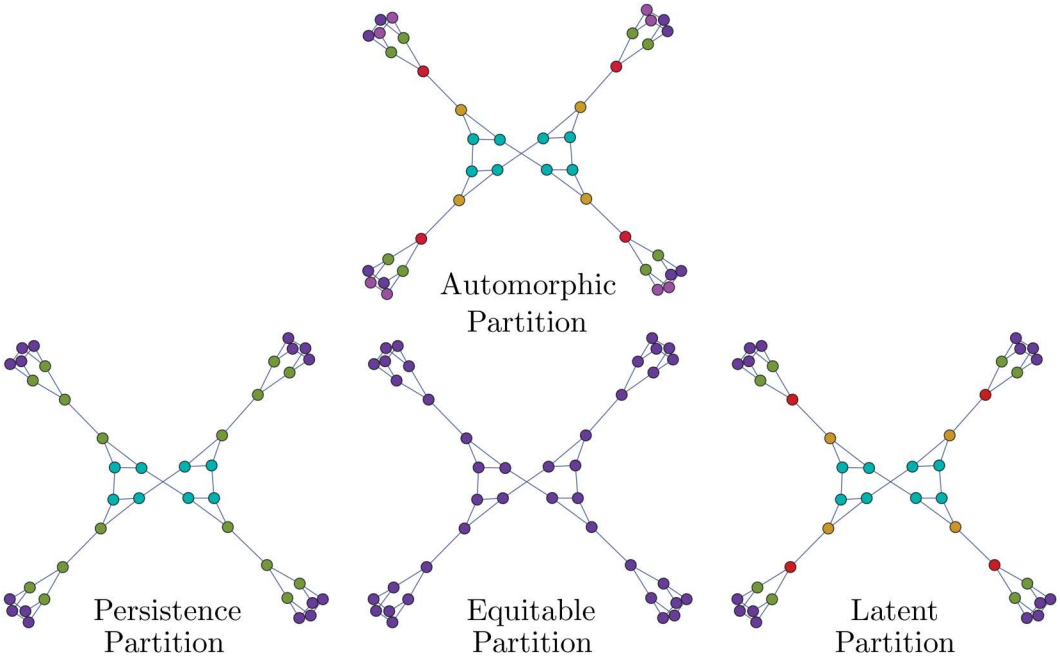

Fig 5. Different partitions of the graph $G=(V,E)$ are shown where vertices in the same cell share the same color for a given partition. Top Center: An automorphic partition $(A_i)_{i=1}^{6}$ is shown. Bottom Left: The persistence partition $(P_i)_{i=1}^{3}$ is shown. Bottom Center: The coarsest equitable partition $(E_i)_{i=1}^{1}$ consisting of a single cell is shown. Bottom Right: The latent partition $(L_i)_{i=1}^{5}$ is shown. The automorphic partition is a refinement of each of the persistence, equitable, and latent partition, respectively (cf. Theorem 1).

https://doi.org/10.1371/journal.pcsy.0000109.g005

For example, the structural partitions of the graphs in Figs 5 and 6 are trivial: each cell contains exactly one vertex. Consequently, these structural partitions refine every other partition on their respective graphs. In contrast, in Fig 8 (the Zachary Karate Club network), the structural partition contains two nontrivial cells, corresponding to the orange and green vertex sets, respectively.

Finally, it is worth noting that regular equivalence is a relaxation of structural equivalence that defines roles by how nodes are similarly connected to equivalent types of other nodes, rather than requiring them to be identically connected [10]. However, we do not consider regular equivalence here, as determining a graph's regular partition can require knowledge of the roles of each vertex in the graph (see [12], for instance). Because this goes beyond the information provided by the graph's topology alone, it falls outside the scope of our analysis.

3.2 Persistence partitions of some synthetic networks

Partitions of real-world networks are often created to detect the community, role, or the core and peripheral structure of a network. Here our goal is to understand whether the PSA finds the partition structure either found or embedded in two network models. The two models we consider are (i) the degree corrected stochastic block model (DCSBM) and (ii) random geometric graphs (RGG).

Definition 13. ***(Degree Corrected Stochastic Block Model)*** *A degree corrected stochastic block model is a model that creates an undirected graph G=(V,E) with planted clusters. The vertex set V is partitioned into communities $C_1, ..., C_k$ where a symmetric matrix $P = [0, 1]^{k\times k}$ determines the probability S_{ij} that vertices $u \in C_i$ and $v \in C_j$ are connected. These probabilities are modified by the parameters $\{\theta_k\}_{k=1}^{n}$ so that there is a directed edge from vertex u to vertex v with probability*

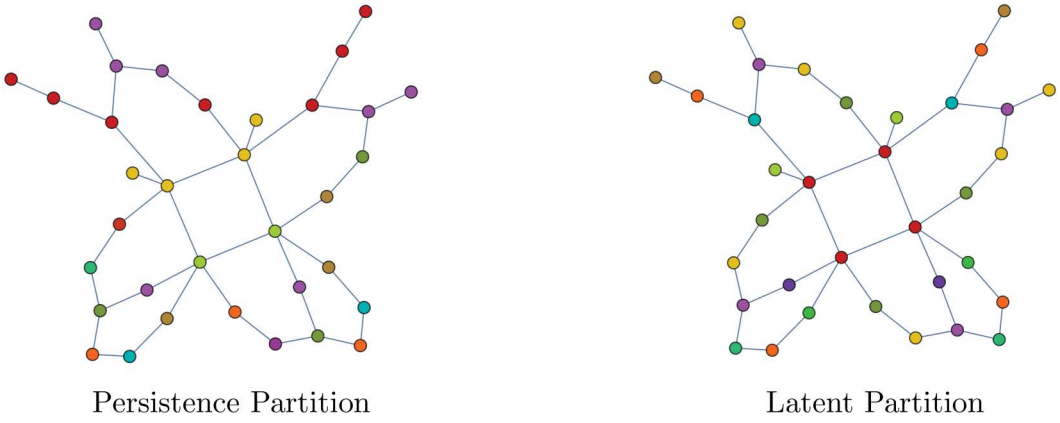

Fig 6. Left: The persistent partition $(P_i)_{i=1}^{11}$ of the graph $G=(V,E)$ is shown where vertices in the same persistent cell have the same color. Right: The latent partition $(L_i)_{i=1}^{11}$ is shown where vertices in the same persistence cell have the same color. Although the two partitions have the same number of cells, the two partitions are incomparable.

https://doi.org/10.1371/journal.pcsy.0000109.g006

$$P(u \to v) = \rho \cdot \theta_u \theta_v S_{ij}$$

where $\rho > 0$ *is choosen such that* $\max_{u,v}(\rho \cdot \theta_u \theta_v S_{ij}) = 1$*. The resulting directed graph is undirected to create the simple graph G=(V,E).*

Example 6. ***(DCSMB and persistence partitions)*** *Consider the stochastic block model network with two communities* C_1 *and* C_2 *where* $|C_1| = |C_2| = 50$*. The probability of edges between vertices of the same community is* $S_{11} = S_{22} = .5$*, and the probability of edges between communities is* $S_{12} = S_{21} = .01$*. We choose each* $\theta_k \in \{1, 2, , \ldots, 100\}$ *uniformly at random for each k=1,2,…,100. A realization of this DCSBM is shown in Fig 7 (left). In contrast, choosing each* $\theta_k = 1$ *results in the standard SMB model. A realization of this SBM is shown in Fig 7 (center) using* $|C_1| = |C_2| = 50$*,* $S_{11} = S_{22} = .2$ *and* $S_{12} = S_{21} = .005$*.*

Running PSA in dimension-2 on this network results in the colored persistence partition indicated in Fig 7 (left, center). What can be seen here, and what happens in general, is that cells of persistence partitions need not belong to either C_1 *or* C_2 *but may have vertices in both.*

To understand the extent to which these persistence cells are aligned with DCSBM and SMB communities, i.e., contained within C_1 *or* C_2*, we compare the adjusted mutual information (AMI) score of the persistence cells against the DCSBM and SMB communities to the AMI scores of one thousand random permutations of the vertex labels against the DCSBM and SMB communities, respectively.*

We find that the persistence cells have an average AMI score of 0.032 for the degree-corrected stochastic block model (DCSBM), indicating that the relationship between persistence cells and the embedded communities in this model is negligible. In contrast, for the standard stochastic block model (SBM), we observe an AMI score of 0.37, reflecting a substantially higher relationship. However, even in this case, the correspondence is only moderate and does not constitute a meaningful match between the persistence cells and the embedded communities.

Next, we consider random geometric graphs, which are defined as follows.

Definition 14. ***(Random Geometric Graph)*** *A random geometric graph (RGG) model is a model that creates an undirected graph G = (V, E) where the vertices are randomly sampled from the uniform distribution spanning an underlying metric space. Vertices are connected by an edge if and only if the distance between them is less than a parameter r>0.*

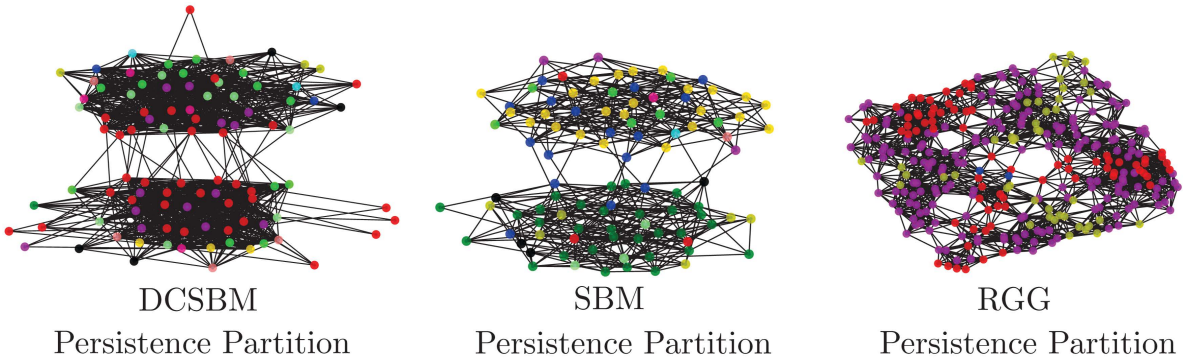

Fig 7. Left: A degree corrected stochastic block model (DCSMB) network on two embedded communities C_1 and C_2 of size fifty each is shown. The network has fourteen distinct cells, indicated by node color, which are not always contained in either C_1 and C_2. Center: A stochastic block model network (SBM) also on two embedded communities C_1 and C_2 of size fifty each is shown (see Example 6 for details). The network has sixteen distinct cells, indicated by node color, which similarly are not always contained in either C_1 and C_2. Right: The four cells in the network's persistence partition of a random geometric graph are shown where cells are indicated by color. The larger three cells are non-localized, spreading across the entire network (see Example 7).

https://doi.org/10.1371/journal.pcsy.0000109.g007

Example 7. ***(RGG and persistence partitions)*** *In Fig 7, an RGG is shown together with its persistence cells for vertices a placed uniformly in a subset of $\mathbb{R}^2$. In this specific realization there are four persistence cells shown as red, purple, yellow, and two blue vertices are spread throughout the network cutting across the embedded communities, i.e., local subgraphs with high modularity. This is in contrast to what is typically seen in the SBM model we considered in Example 6. While the precise reason for this is unknown, this indicates that persistence cells can be quite different from standard communities, and it is an open question as to how the number, size, and distribution of persistence cells depend on the parameters $|V| = n$ and $r > 0$ in an RGG model.*

3.3 Persistence partitions of two real-world networks

Zachary's karate club is a social network that models the social interactions of members of a karate club. This network is of particular interest as the club eventually split, with roughly half of the students following the club instructor and the other half following the club president [44] (see S2 Text and S2 Data). This split is recovered almost exactly by maximizing the modularity over all two-community partitions (see Fig 8, top left). We refer to these as the instructor community and president community, which are shown in red and blue, respectively.

The persistence partition of the unweighted Zachary's karate club also recovers this split to some degree (see Fig 8, right). In particular, all of the network's twelve persistence cells except one belong to either the instructor's community or the president's community. We refer to the exception as the *mixed community*, which is shown in purple. The president and instructor each form their own persistence cell, and the other cells form the peripheral structure of the network on the side of the instructor and/or president. In this way of thinking, each individual performs a specific type of role within the network relative to the instructor and president. For instance, the individuals in the dark green community interact only with the president and the second most well-connected individual in the president's community by degree.

To provide a point of comparison, we analyze the network's core–periphery structure (see, e.g., [16,45,46]). Detection is performed using the method proposed in [18], which is based on isospectral reduction (see Definition 11). Applying this method using the criterion that selects all vertices except those with the smallest degree centrality yields the core–periphery structure shown in Fig 8 (bottom right). This procedure identifies seven hierarchical levels. This includes the core followed by successive peripheral layers, corresponding to the vertices colored red, yellow, purple, orange, green, indigo, and dark blue. The innermost core (red vertices) contains both the club's instructor and president, along with several

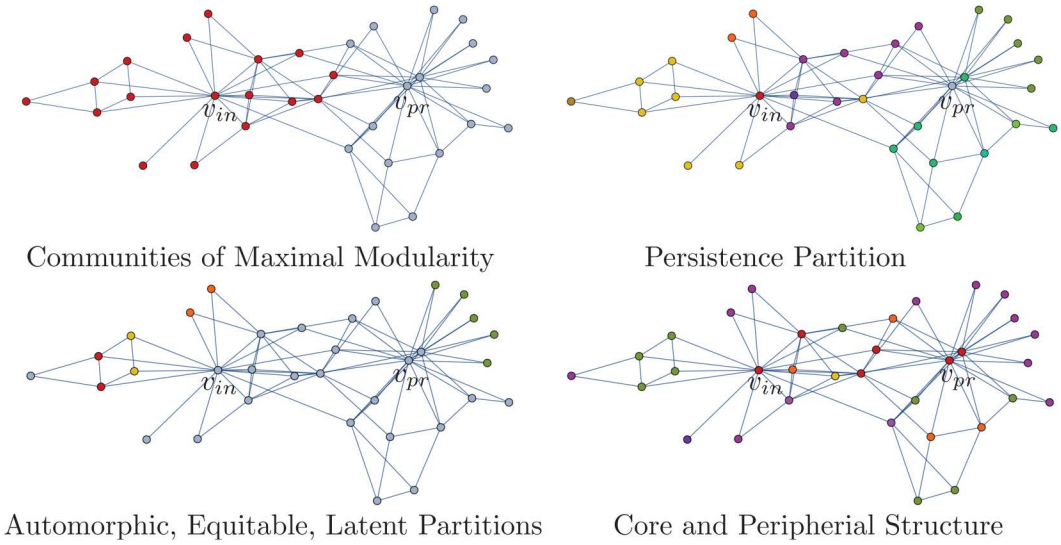

Fig 8. Zachary's karate club is shown both left and right, top and bottom with the club's instructor and president indicated by v_{in} and v_{pr}, respectively. Top Left: Maximizing modularity over all two-community partitions results in those shown in red and blue. Top Right: The twelve persistence cells of Zachary's karate club are shown, indicated by color. Bottom Left: The automorphic, coarsest equitable, and latent partitions of the network are shown; in this case, all three coincide. Colors denote the cells of the partition, with the exception that each blue vertex forms its own singleton cell. Bottom Right: The core-periphery structure of the network is shown created using isospectral reductions.

https://doi.org/10.1371/journal.pcsy.0000109.g008

members occupying paths between them. As one moves outward, the peripheral layers expand from these central vertices. This pattern differs substantially from that of the network's persistence partition resulting in two partitions are incomparable. The key difference is that the core–periphery structure is less refined, merging the orange and green vertices of the automorphic/equitable/latent partition, whereas these are separated in the persistence partition.

Currently, it is unknown whether persistence partitions correspond more to specific roles in real-world networks, to standard notions of communities, or are better suited for describing core and peripheral structures (cf. the New York taxicab network, SMB and RGG models, and the Zachary karate club, respectively).

As a second example, the New York taxicabs network $G=(V,E)$ consists of $|V|=265$ taxi zones or neighborhoods in New York City where there is an edge $\{i,j\} \in E$ between neighborhoods i and j if there was a taxicab that picked up a passenger in neighborhood i and dropped them off in neighborhood j [47] (see S3 Text and S3 Data). Here we consider the edges to be undirected and unweighted, although, as mentioned in the introduction, both directionality and edge weight could be incorporated into our persistence surface algorithm. For simplicity, we only consider the undirected and unweighted version of this network, ignoring the directionality and frequency of travel.

The persistence partition of the New York taxicab network is shown in Fig 9, where color is used to indicate each of the cells. The network itself is not shown, but rather the neighborhood map of New York City. This is to emphasize the fact that persistence cells are often not spatially connected but are spread throughout the city's five boroughs. That is, there are neighborhoods that are distant from one another, but nevertheless see the same types of traffic dynamics. We note that this interpretation suggests that persistence partitions represent specific transportation roles, as opposed to the more classical notion of communities, such as the city's boroughs.

Similarly to Zachary's karate club, the taxicab's persistence cells do not appear to represent standard communities, rather specific types of communities that are more aligned with roles or functions, each determined by structure rather than modularity. In the context of the New York City taxicab network, this distinction may be useful in the process of designing and building city roads and other traffic-related infrastructure, as topologically similar regions may experience

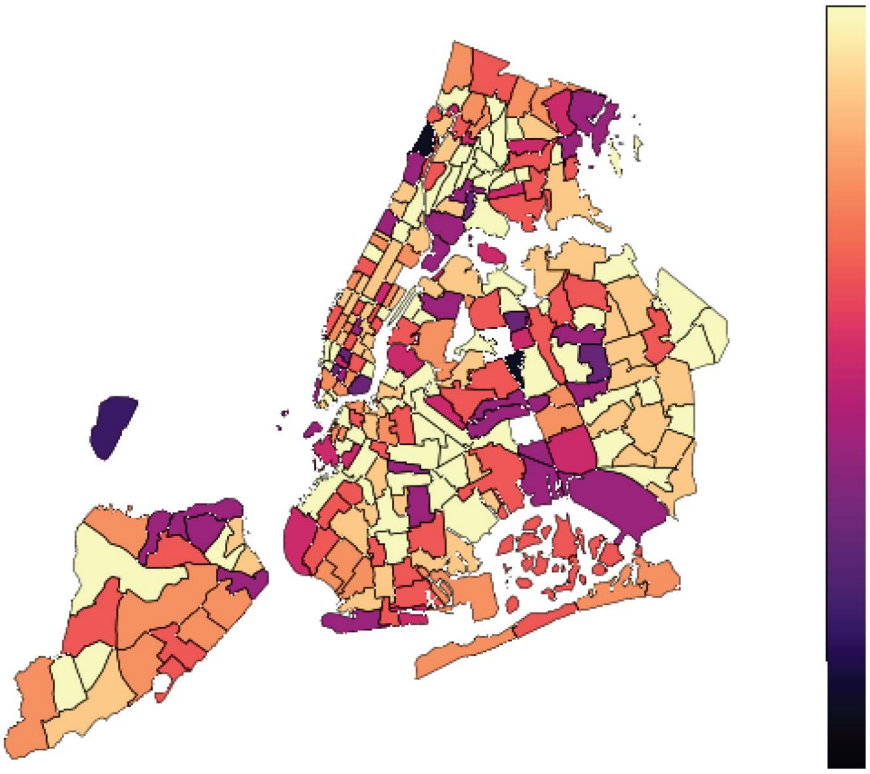

Fig 9. The persistence partition of the New York taxicab network is shown where color is used to indicate each of the cell. The network consists of vertices representing city neighborhoods and edges indicating whether taxi cabs move passengers from one neighborhood to another. The network itself is not shown but rather the neighborhood map of New York City is shown to scale. The majority of the ten persistence cells are spread throughout each of the five New York City boroughs. The base layer of the map can be found at https://d37ci6vzurychx.cloudfront.net/misc/taxi_zones.zip. The dataset is publicly available under the New York City Open Data Terms of Use: https://opendata.cityofnewyork.us/overview/#termsofuse.

https://doi.org/10.1371/journal.pcsy.0000109.g009

similar traffic patterns. If a particular infrastructure works well in neighborhood i, we might predict that the same infrastructure may be useful in region j if neighborhoods i and j are in the same persistence cell.

4 Discussion

In this paper we propose the concept of a persistence surface, which can be constructed for each vertex $v \in V$ of a graph $G=(V,E)$. By grouping vertices according to their persistence surfaces, we create a partition of the graph's vertices whose elements or cells form the persistence partition of the graph.

Currently, it is not known how a network's persistence partition is relate to its community structure that is typically studied in network science. The two appear correlated at times (see the SBM and Zachary's karate club examples) and quite different at others (see the RGG example and New York taxicab network). Similarly, it is unknown whether persistence partitions are a better approximation of the roles played in a network or a useful method for determining a network's core and periphery structure.

With respect to other partitions of a network's vertices, we show that automorphic partition always refines the persistence partition, as well as the two new partitions introduced here; equitable and latent partitions. Although persistence, equitable, and latent partitions may sometimes refine one another and may be incomparable in other cases, the precise relationships among these three partition types remain unknown.

It is likewise unclear how these new network partitions compare with recent advances in algorithmic network partitioning, such as algorithmic probability methods [48] and low-rank, random-walk–based, and deep-learning–based network embedding methods [11]. A thorough comparison lies beyond the scope of the present work, but we intend to examine these relationships in a subsequent paper. At present, whether these modern techniques identify structures analogous to those uncovered by persistence partitions remains an open question.

Supporting information

S1 Text. Documentation for S1 Data. Text file describing the structure, variables, and formatting of the S1 Data for the Gundangborn network dataset.
(TXT)

S1 Data. Gundangborn network dataset. CSV file containing the Gundangborn network data.
(CSV)

S2 Text. Documentation for S2 Data. Text file describing the structure of the S2 Data for the Zachary Karate Club dataset.
(TXT)

S2 Data. Zachary Karate Club network dataset. CSV file containing the Zachary Karate Club network data.
(CSV)

S3 Text. Documentation for S3 Data. Text file describing the structure of the New York Taxi Network Dataset.
(TXT)

S3 Data. Taxi network dataset. Parquet file containing the Nwe York City Taxi network data.
(PARQUET)

Acknowledgments

Code availability: Code to determine the persistence communities of a network can be found at https://github.com/AbigailJ32/PS-community-detection.

Author contributions

Conceptualization: Benjamin Z. Webb, Abigail Jenkins, Zachary M. Boyd.

Data curation: Abigail Jenkins, Taylor Gledhill.

Formal analysis: Benjamin Z. Webb, Abigail Jenkins, Nick Callor.

Funding acquisition: Benjamin Z. Webb.

Investigation: Benjamin Z. Webb, Abigail Jenkins, Nick Callor, Zachary M. Boyd, Taylor Gledhill, Raelynn Wonnacott.

Methodology: Abigail Jenkins, Zachary M. Boyd, Taylor Gledhill, Raelynn Wonnacott.

Supervision: Benjamin Z. Webb, Nick Callor, Zachary M. Boyd.

Validation: Benjamin Z. Webb, Abigail Jenkins, Nick Callor, Zachary M. Boyd, Raelynn Wonnacott.

Visualization: Abigail Jenkins, Nick Callor, Raelynn Wonnacott.

Writing – original draft: Benjamin Z. Webb, Abigail Jenkins, Nick Callor, Zachary M. Boyd.

Writing – review & editing: Benjamin Z. Webb, Abigail Jenkins, Nick Callor, Zachary M. Boyd, Taylor Gledhill, Raelynn Wonnacott.

References

1. Chen M, Kuzmin K, Szymanski BK. Community Detection via Maximization of Modularity and Its Variants. IEEE Trans Comput Soc Syst. 2014;1(1):46–65. https://doi.org/10.1109/tcss.2014.2307458
2. Newman MEJ. Spectral methods for community detection and graph partitioning. Phys Rev E Stat Nonlin Soft Matter Phys. 2013;88(4):042822. https://doi.org/10.1103/PhysRevE.88.042822 PMID: 24229240
3. Wang T, Yin L, Wang X. A community detection method based on local similarity and degree clustering information. Physica A: Statistical Mechanics and its Applications. 2018;490:1344–54. https://doi.org/10.1016/j.physa.2017.08.090
4. Lee C, Wilkinson DJ. A review of stochastic block models and extensions for graph clustering. Appl Netw Sci. 2019;4(1). https://doi.org/10.1007/s41109-019-0232-2
5. Duchemin Q, De Castro Y. Random geometric graph: Some recent developments and perspectives. In: Adamczak R, Gozlan N, Lounici K, Madiman M. High Dimensional Probability IX. Springer International Publishing. 2023. 347–92.
6. Scripps J, Tang PN, Esfahanian AH. Node roles and community structure in networks. In: Proceedings of the ACM. New York, NY, USA: Association for Computing Machinery. 2007.
7. Coux C, Rader R, Bartomeus I, Tylianakis JM. Linking species functional roles to their network roles. Ecol Lett. 2016;19(7):762–70. https://doi.org/10.1111/ele.12612 PMID: 27169359
8. Peel L. Active discovery of network roles for predicting the classes of network nodes. Jcomplexnetw. 2014;3(3):431–49. https://doi.org/10.1093/comnet/cnu043
9. Sailer L. Structural equivalence: Meaning and definition, computation and application. Social Networks. 1978;1(1):73–90.
10. White DR, Reitz KP. Graph and semigroup homomorphisms on networks of relations. Social Networks. 1983;5(2):193–234. https://doi.org/10.1016/0378-8733(83)90025-4
11. Jiao P, Guo X, Pan T, Zhang W, Pei Y, Pan L. A Survey on Role-Oriented Network Embedding. IEEE Transactions on Big Data. 2022;8(4).
12. Park H, Neville J. Role Equivalence Attention for Label Propagation in Graph Neural Networks. Advances in Knowledge Discovery and Data Mining. 2020.
13. Lorrain F, White HC. Structural equivalence of individuals in social networks. The Journal of Mathematical Sociology. 1971;1(1):49–80. https://doi.org/10.1080/0022250x.1971.9989788
14. da Silva MR, Hongwu Ma, Zeng AP. Centrality, Network Capacity, and Modularity as Parameters to Analyze the Core-Periphery Structure in Metabolic Networks. Proc IEEE. 2008;96(8):1411–20. https://doi.org/10.1109/jproc.2008.925418
15. Rossa FD, Dercole F, Piccardi C. Profiling core-periphery network structure by random walkers. Sci Rep. 2013;3:1467. https://doi.org/10.1038/srep01467 PMID: 23507984
16. Ansari I, Yadav P, Sahni N. Uncovering the hidden core–periphery structure in hyperbolic networks. Physical Review E. 2025;112(3):034311. https://doi.org/10.1103/PhysRevE.112.034311
17. Bunimovich L, Smith D, Webb B. Finding Hidden Structures, Hierarchies, and Cores in Networks via Isospectral Reduction. Applied Mathematics and Nonlinear Sciences. 2019;4(1):231–54. https://doi.org/10.2478/amns.2019.1.00021
18. Bunimovich LA, Wang C-J, Chae S, Webb BZ. Uncovering Hierarchical Structure in Social Networks Using Isospectral Reductions. In: 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), 2018. 1199–206. https://doi.org/10.1109/asonam.2018.8508506
19. Otter N, Porter MA, Tillmann U, Grindrod P, Harrington HA. A roadmap for the computation of persistent homology. EPJ Data Sci. 2017;6(1):17. https://doi.org/10.1140/epjds/s13688-017-0109-5 PMID: 32025466
20. Li M, Zhang T, Chen Y, Smola AJ. Persistent homology: An introduction and a new text representation for natural language processing. In: Proceedings of the International Joint Conference on Artificial Intelligence, 2014. 1953–9.
21. de Silva V, Ghrist R. Coverage in sensor networks via persistent homology. Algorithms for Sensor and Ad Hoc Networks. Springer. 2007. 273–87.
22. Robins V, Saadatfar M, Delgado-Friedrichs O, Sheppard AP. Percolating length scales from topological persistence analysis of micro-CT images of porous materials. Water Resources Research. 2016;52(1):315–29. https://doi.org/10.1002/2015wr017937
23. Boyd ZM, Callor N, Gledhill T, Jenkins A, Snellman R, Webb B, et al. The persistent homology of genealogical networks. Appl Netw Sci. 2023;8(1):15. https://doi.org/10.1007/s41109-023-00538-7 PMID: 36852178
24. Petri G, Expert P, Turkheimer F, Carhart-Harris R, Nutt D, Hellyer PJ, et al. Homological scaffolds of brain functional networks. J R Soc Interface. 2014;11(101):20140873. https://doi.org/10.1098/rsif.2014.0873 PMID: 25401177
25. Hatcher A. Algebraic Topology. Cambridge University Press. 2002.
26. Cohen-Steiner D, Edelsbrunner H, Harer J. Stability of Persistence Diagrams. Discrete Comput Geom. 2006;37(1):103–20. https://doi.org/10.1007/s00454-006-1276-5
27. McCarthy FD, McArthur M. Gundangborn 1948 AU02 dataset. Kinsources. 2010. https://www.kinsources.net/kidarep/dataset-81-gundangborn-1948-au02.xhtml
28. Ripser Developers. Ripser Python package. 2021. https://anaconda.org/conda-forge/ripser

29. MacArthur BD, Sánchez-García RJ, Anderson JW. Symmetry in complex networks. Discrete Applied Mathematics. 2008;156(18):3525–31. https://doi.org/10.1016/j.dam.2008.04.008
30. Scholkemper M, Schaub MT. An Optimization-based Approach To Node Role Discovery in Networks: Approximating Equitable Partitions. In: Advances in Neural Information Processing Systems 36, 2023. 71358–74. https://doi.org/10.52202/075280-3125
31. Smith D, Webb B. Hidden symmetries in real and theoretical networks. Physica A: Statistical Mechanics and its Applications. 2019;514:855–67. https://doi.org/10.1016/j.physa.2018.09.131
32. Brouwer A, Haemers W. Spectra of graphs. Springer. 2012.
33. Pecora LM, Carroll TL. Master Stability Functions for Synchronized Coupled Systems. Phys Rev Lett. 1998;80(10):2109–12. https://doi.org/10.1103/physrevlett.80.2109
34. Sorrentino F, Pecora LM, Hagerstrom AM, Murphy TE, Roy R. Complete characterization of the stability of cluster synchronization in complex dynamical networks. Science Advances. 2016;2(4):e1501737. https://doi.org/10.1126/sciadv.1501737
35. Hannesson E, Sellers J, Walker E, Webb B. Network specialization: A topological mechanism for the emergence of cluster synchronization. Physica A: Statistical Mechanics and its Applications. 2022;600:127496. https://doi.org/10.1016/j.physa.2022.127496
36. King A, Smith D, Webb B. Minimal specialization: Coevolution of network structure and dynamics. Physica A: Statistical Mechanics and its Applications. 2024;651:130000. https://doi.org/10.1016/j.physa.2024.130000
37. Scheinerman E, Ullman D. Fractional graph theory: A rational approach to the theory of graphs with a foreword by Claude Berge. John Wiley. 1997.
38. Bunimovich LA, Webb BZ. Isospectral graph transformations, spectral equivalence, and global stability of dynamical networks. Nonlinearity. 2011;25(1):211–54. https://doi.org/10.1088/0951-7715/25/1/211
39. Bunimovich LA, Webb BZ. Isospectral graph reductions and improved estimates of matrices' spectra. Linear Algebra and its Applications. 2012;437(7):1429–57. https://doi.org/10.1016/j.laa.2012.04.031
40. Carter C, Murri J, Reber D, Webb B. A simple stability criterion for dynamical systems with stochastic switching and/or stochastic time-delays. Nonlinearity. 2022;35(12).
41. Balls-Barker B, Webb B. Link prediction in networks using effective transitions. Linear Algebra and its Applications. 2020;599:79–104. https://doi.org/10.1016/j.laa.2020.04.006
42. Kempton M, Sinkovic J, Smith D, Webb B. Characterizing cospectral vertices via isospectral reduction. Linear Algebra and its Applications. 2020;594:226–48. https://doi.org/10.1016/j.laa.2020.02.020
43. Webb B, Bunimovich L. Isospectral transformations: A new approach to analyzing multidimensional systems and networks. New York: Springer-Verlag. 2014.
44. Zachary WW. An Information Flow Model for Conflict and Fission in Small Groups. Journal of Anthropological Research. 1977;33(4):452–73. https://doi.org/10.1086/jar.33.4.3629752
45. Borgatti SP, Everett MG. Models of core/periphery structures. Social Networks. 2000;21(4):375–95. https://doi.org/10.1016/s0378-8733(99)00019-2
46. Rombach MP, Porter MA, Fowler JH, Mucha PJ. Core-Periphery Structure in Networks. SIAM J Appl Math. 2014;74(1):167–90. https://doi.org/10.1137/120881683
47. New York City Taxi Cab Dataset. https://www.nyc.gov/site/tlc/about/tlc-trip-record-data.page
48. Zenil H, Kiani NA, Zea AA, Tegnér J. Causal deconvolution by algorithmic generative models. Nat Mach Intell. 2019;1(1):58–66. https://doi.org/10.1038/s42256-018-0005-0