

Hodge-Decomposition of Brain Networks

D. Vijay Anand¹, Soumya Das², Moo K. Chung²

¹University of Exeter, United Kingdom, ²University of Wisconsin, Madison, USA

Abstract. We propose to analyze brain networks by decomposing them into three orthogonal components through the Hodge decomposition. We propose to quantify the magnitude and relative strength of each components. We performed extensive simulation studies with the known ground truth. The Hodge decomposition is then applied to the human brain networks obtained from the resting state functional magnetic resonance imaging study. Our study indicates that the components of the Hodge decomposition contain topological features that provide statistically significant difference between the male and female brain networks.

1 Introduction

Understanding the topological patterns in the neuronal activity of the human brain is fundamental in understanding the brain function and disease prognosis. Many brain disorders such as schizophrenia, epilepsy, autism, and Alzheimer’s disease (AD) exhibit abnormal patterns in the brain activity as evidenced from numerous functional magnetic resonance imaging (fMRI) studies [1,2]. Often these brain images are processed and represented as networks and analyzed using the graph theory [3,4,5]. The graph models provide quantitative measures such as centrality [5,6], community detections [7] and hubs [8,9]. While the graph theory approaches were successful in incorporating the pairwise (dyadic) relations between different brain regions, it often fail to capture the coherent behaviors such as the co-activation of brain areas and co-firing of neurons [10]. The inherent *dyadic* assumption in the graph limits the types of neural structure and function that the graphs can model [10,11]. The brain network models built on top of graphs cannot encode higher order interactions such as three- and four-way interactions, beyond pairwise connectivity *without* additional analysis [10]. To this end, encoding and interpreting brain networks beyond pairwise interactions is of utmost importance.

Recently, there is a plethora of researches across different disciplines that emphasize the need for higher-order networks [11,10,12,13]. Despite its prevalence, the studies considering the higher order interactions pertaining to brain networks are very limited [14]. Understanding the higher order interactions of the brain regions is crucial in order to model the structural and functional organization of the brain networks [15]. The topological data analysis (TDA) based techniques provide the necessary tools to probe the geometric and topological structures of these higher order systems [16,17]. The mathematical construct used for this purpose is the *simplicial complex*, which contain basic building blocks referred to as simplices: nodes (0-simplices), edges (1-simplices), triangles (2-simplices) and tetrahedrons (3-simplices). These simplices systematically encode higher order interactions [10,18]. The persistent homology (PH), one of major TDA techniques deeply rooted in simplicial complexes, enables coherent network representation at different spatial resolutions in obtaining higher order topological features [19,20]. The PH based approaches are widely used to understand the brain imaging data [21,22,14,23,24]. The core idea of PH is to generate a series of nested algebraic structures over multiple scales through a filtration process [19]. The PH allows to quantify the multiscale topological characteristics through topological invariants [25]. Recently, the Hodge theory based approaches have been successful in analyzing complex networks since they provide both the topological invariants and the spectral characteristics that can uncover the deep underlying of physical mechanisms hidden in data. [26,27,28].

The Hodge theory offers a unified framework that constitutes both simplicial homology and spectral geometry. In the last decade, it has become an important tool in topological data analysis, particularly in the study of algebraic cycles [26], higher order diffusion [29], denoising [13], and consensus dynamics [30], and statistical and machine learning [28]. However, the method is not yet adapted for analyzing brain networks. The Hodge Laplacian is a generalization of the graph Laplacian into its higher dimensional counterpart on simplicial complexes [31]. The topological features of the underlying structure such as cycles and community clusters can be extracted from the spectral decomposition of the Hodge Laplacians [32]. On the other hand, the Hodge decomposition provides an efficient representation for the data or signals defined on simplices and establishes relationship between simplices of different dimensions [33]. The Hodge decomposition of functional data on a p -simplex can be decomposed such that the subspaces contain information from $p - 1$ and $p + 1$ simplices. In particular, the Hodge decomposition on 1-simplices (edges) have been studied as a natural way of modeling flows in networks. The functional data defined on the edges with orientation can be expressed as a skew symmetric matrix and is referred to as edge flows [28]. The Hodge decomposition of the edge flow results in three orthogonal components called the gradient, curl and harmonic flows. These components carry topological information corresponding to the upper and lower simplex dimensions [27]. Functional data defined on a 1-simplex (edges) has components that are formed from the functions defined on the 0-simplex (nodes) and 2-simplex (triangles).

In this study, we aim to disclose how the three components of the Hodge decomposition can be used to explain the interplay between higher-order interactions and topology. The central idea is to project the data defined on the edges into other simplicial dimensions using the Hodge decomposition such that each component encodes specific information about the topology of the network. The gradient flows corresponds to flows induced by gradients on the nodes, the curl flow corresponds to flows that curl around triangles and the harmonic flow corresponds to both curl-free and gradient-free flows [30,34]. The harmonic flow component is related to the loop structures and captures topological signatures such as the cycles in a network. The curl and the harmonic flows are cyclic in nature and together they are referred as *loop flows*. Further, the gradient flow is acyclic and hence we refer them as *non-loop flows* [27].

We project functional data defined on the edges onto the nodes and triangles and compute the gradient and curl flows respectively. The relative strength of these flows are quantified using measures such as the gradient ratio, curl ratio and loop ratio [35]. We design a Wasserstein distance based statistical inference procedure to examine the topological similarity and dissimilarity of the loop flows and non-loop flows and assess their capability in discriminating networks. We also introduce an efficient algorithm to identify and extract 1-cycles (loops) using the curl and harmonic components. We then use the algebraically independent 1-cycles as basis and proceed to compute the most discriminating cycles using the maximum gap statistic. We also perform extensive simulation studies with the ground truth to validate our claims. The proposed methods are further applied to real brain network data obtained from a functional magnetic resonance imaging study in discriminating the flow patterns of the female and brain male networks.

2 Method

The Helmholtz-Hodge decomposition (HHD), commonly referred to as the Hodge decomposition is a generalization of the Helmholtz decomposition. HHD is often used in science and engineering applications establishes that any vector field \mathbf{V} can be described as a sum of three orthogonal components [36]:

$$\mathbf{V} = \nabla s + \nabla \times \phi + \mathbf{h}.$$

The first term ∇s is the gradient of a scalar potential field s . The second term $\nabla \times \phi$ is the curl of a vector field ϕ . The last term \mathbf{h} is a harmonic vector field. The vector field ψ

is known as the vector potential and the curl component of the decomposition is divergence free. Once we compute the functions s and ϕ for a given vector field we can compute the harmonic term. More details on the HHD on vector fields can be found in [37,38]. A natural extension of this decomposition applied to simplicial complexes is possible via the discrete Hodge theory or combinatorial Hodge theory.

2.1 Graphs as simplicial complexes

A simplicial complex is a collection of simplices such as nodes (0–simplices), edges (1-simplices), triangles (2-simplices), a tetrahedron (3-simplices) and higher dimensional counterparts. More precisely, a simplicial complex K is a finite set of simplexes that satisfy two conditions: any face of a simplex from K is also in K ; the intersection of any two simplexes in K is either empty or a shared face [16]. Assuming p number of nodes v_0, v_1, \dots, v_{p-1} , a p -simplex can be written as $\sigma_p = [v_0, v_1, \dots, v_{p-1}]$. A simplicial complex consisting of up to p -simplexes is called a p -skeleton. Thus, the graphs are 1-skeletons [39]. A simplicial complex can be viewed as the higher dimensional generalization of a graph [40,16,19].

Graph filtration Consider weighted graph $G = (V, w)$ with edge weights $w = w_{ij}$ between nodes v_i and v_j . A binary graph $G_\epsilon = (V, w_\epsilon)$ is a graph having node set V and the binary edge weights $w_\epsilon = (w_{\epsilon,ij})$ written as

$$w_{\epsilon,ij} = \begin{cases} 1 & \text{if } w_{ij} > \epsilon, \\ 0 & \text{otherwise.} \end{cases}$$

A graph filtration of G is defined as the collection of nested binary graphs

$$G_{\epsilon_0} \supset G_{\epsilon_1} \supset \dots \supset G_{\epsilon_k}$$

with filtration values $\epsilon_0 < \epsilon_1 < \dots < \epsilon_k$ [41,42]. A unique filtration can be obtained by using the sorted the edge weights

$$\min_{i,j} w_{ij} = w_{(1)} < w_{(2)} < \dots < w_{(q)} = \max_{ij} w_{ij}$$

as filtration values. Given a complete weighted graph with p nodes, there are $q = \binom{p}{2} = p(p-1)/2$ edge weights.

A graph filtration is basically built starting from a complete graph $G_{-\infty}$ and proceeding towards node set $G_{w_{(q)}}$. This is achieved by sequentially removing one edge at a time from the complete graph till all the edges are removed.

Birth-death decomposition In conventional topological data analysis applied to simplicial complexes, the number of connected components and the number of cycles are called as the topological invariants or the Betti numbers [43]. The topological invariants in a network are usually summarized as barcodes [25] and are usually represented as pairs of birth and death values of topological features. The set of birth and death values represent the filtration values at which the features are born and destroyed respectively. Since they always come in pairs they are easily represented as 2D scatter points in persistent diagrams [44]. However, the use of graph filtration simplifies such pairings 1D scatter points.

During the graph filtration, when an edge is deleted, either a new connected component is born or a cycle is destroyed. However these events are exclusive to each other and does not happen at the same time [44,13]. Once a component is born it continues to persist and never dies. Therefore its death values are ∞ and can be ignored [44,13]. The set of filtration values that corresponds to the birth of a component is referred as the birth set

$$B(G) : b_1 < b_2 < \dots < b_{q_0},$$

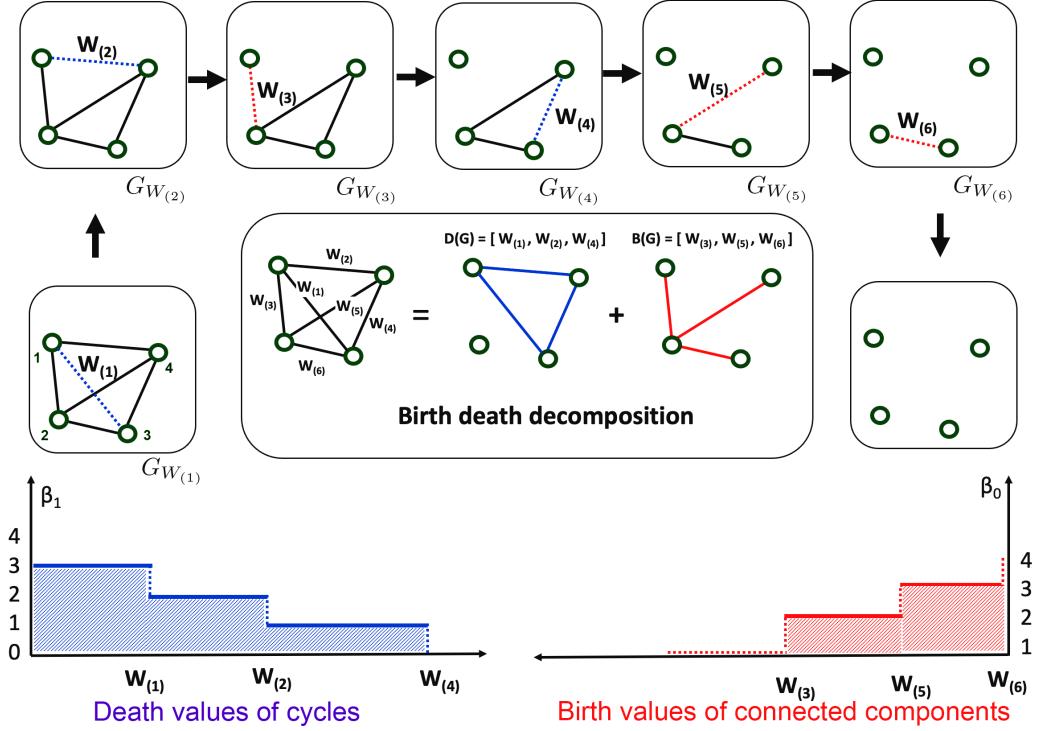


Fig. 1: Illustration of the birth-death decomposition, which portions the edge set into two disjoint birth and death sets. The birth set (red) corresponds to a maximum spanning tree (MST) while the remaining edges (blue) form the death set.

where $q_0 = p - 1$ is the cardinality of the birth set [44]. $B(G)$ characterizes the 0D homology of the graph filtration.

Similarly, all the loops (1-cycles) are considered born at $-\infty$. During the graph filtration, when a cycle is destroyed, we associate that edge weight as the death value. Since the birth and death are exclusive events, the total number of death values of 1-cycles is

$$q_0 = q - q_1 = (p - 1)(p - 2)/2.$$

The set of increasing death values is written as

$$D(G) : d_1 < d_2 < \dots < d_{q_1}.$$

$D(G)$ characterizes the 1D homology of the graph filtration.

Subsequently, the edge set is decomposed into two parts with one set containing the edges that create a component and the other set containing edges that destroy cycles. This can be formally stated as Theorem 1 [44,13,45].

Theorem 1 (Birth-death decomposition). *The set of 0D birth values $B(G)$ and 1D death values $D(G)$ partition the edge weight set W such that $W = B(G) \cup D(G)$ with $B(G) \cap D(G) = \emptyset$. The cardinalities of $B(G)$ and $D(G)$ are $p - 1$ and $(p - 1)(p - 2)/2$ respectively.*

Since the graph filtration is performed on the sorted edge weights, the sequential removal of edges from a complete graph results in a birth set which corresponds to a maximum spanning tree (MST) [44,13]. Thus the computation of birth set $B(G)$ is equivalent to the finding MST of G , which can be easily done using Kruskal's or Prim's algorithms [42,44]. The death set $D(G)$ is subsequently given as edges not in $B(G)$. Thus, the birth and death sets an

be computed efficiently in $\mathcal{O}(q \log q)$. Figure 2 displays the birth-death decompositions of a complete graph with different edge weights. If edge weights are different, the decomposition produces topologically different birth and death sets.

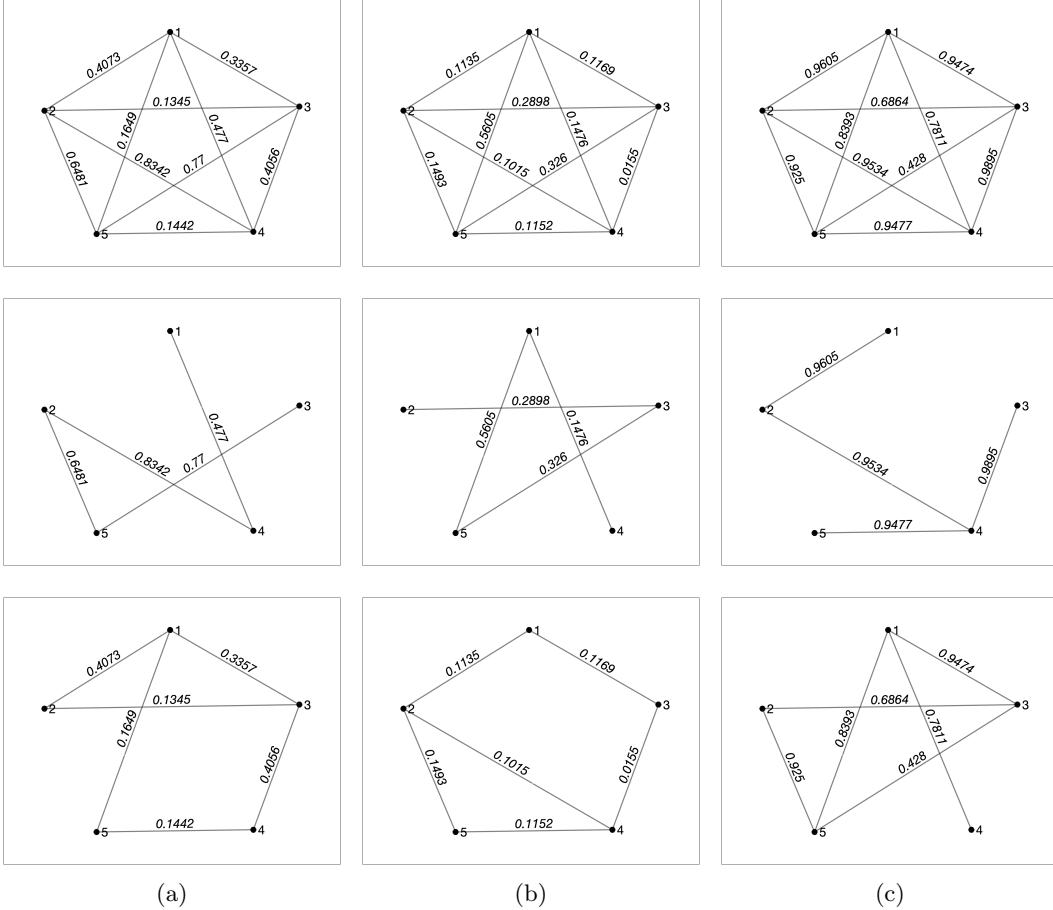


Fig. 2: The birth-death decomposition of three weighted graphs (top) producing topologically different birth set (middle) and death sets (bottom).

Wasserstein distances The topological similarity or dissimilarity between the networks can be computed using the Wasserstein distance between persistent diagrams [46,47,48,44]. Wasserstein distance has been previously used for shape modeling [49,50,51,52], brain volume registration [53] and cognitive prediction [54]. However, it was never applied to the Hodge decomposition and it is unclear how it will perform for the decompostion. We will provide the extensive performance in the Hodge decomposition in the validation studies later.

Let $\Omega = (V^\Omega, w^\Omega)$ and $\Psi = (V^\Psi, w^\Psi)$ be two given networks with p nodes. Their persistent diagrams denoted as P_Ω and P_Ψ are expressed in terms of pairing of birth values (x -coordinates) and death values (y -coordinates). If we model the scatter points in the persistent diagrams P_Ω and P_Ψ using the Dirac-delta functions [13], the r -Wasserstein distance on persistent diagrams is given by

$$\mathfrak{L}_r(P_\Omega, P_\Psi) = \inf_{\tau: P_\Omega \rightarrow P_\Psi} \left(\sum_{x \in P_\Omega} \|x - \tau(x)\|^r \right)^{1/r}$$

over every possible bijection τ between P_Ω and P_Ψ [48]. The ∞ -Wasserstein distance given as the limit of r -Wasserstein distance $\mathfrak{L}_r(P_\Omega, P_\Psi) \rightarrow \mathfrak{L}_\infty(P_\Omega, P_\Psi)$ as $r \rightarrow \infty$ is

$$\mathfrak{L}_\infty(P_\Omega, P_\Psi) = \inf_{\tau: P_\Omega \rightarrow P_\Psi} \max_{x \in P_\Omega} \|x - \tau(x)\|.$$

Generally the Wasserstein distance is computed using the Kuhn-Munkres approach and Hungarian algorithm, which takes quadratic run time [55,56,57]. However, in graph filtrations, the persistent diagrams are reduced to 1D scatter points and the bijection τ is simply given by matching sorted birth and death values reducing the computation to $\mathcal{O}(q \log q)$ run time [44,58].

Theorem 2.

$$\begin{aligned}\mathfrak{L}_r(P_\Omega, P_\Psi) &= \left(\sum_{x \in P_\Omega} \|x - \tau^*(x)\|^r \right)^{1/r} \\ \mathfrak{L}_\infty(P_\Omega, P_\Psi) &= \max_{x \in P_\Omega} |x - \tau^*(x)|,\end{aligned}$$

where τ^* maps the i -th smallest value in P_Ω to the i -th smallest value in P_Ψ for all i .

The minimizations in Theorem 2 is obtained through the *nonlinear rearrangement inequality* [58, p.81]

$$\sum_{x \in P_\Omega} |x - \tau(x)|^r \geq \sum_{x \in P_\Omega} |x - \tau^*(x)|^r$$

for any bijection τ [59]. When we apply Theorem 2 to graph filtrations, we can analytically express the Wasserstein distances in terms of matched birth and death values.

Theorem 3. For r -Wasserstein distances for 0D and 1D homology on graph flirtations are given by

$$\mathfrak{L}_r^b(P_\Omega, P_\Psi) = \left(\sum_{i=1}^{q_0} |b_i^\Omega - b_i^\Psi|^r \right)^{1/r}, \quad \mathfrak{L}_r^d(P_\Omega, P_\Psi) = \left(\sum_{i=1}^{q_1} |d_i^\Omega - d_i^\Psi|^r \right)^{1/r},$$

where b_i^Ω and b_i^Ψ are the i -th smallest birth values of Ω and Ψ . Similarly, ∞ -Wasserstein distance for 0D and 1D homology on graph flirtations are given by

$$\mathfrak{L}_\infty^b(P_\Omega, P_\Psi) = \max_{1 \leq i \leq q_0} |b_i^\Omega - b_i^\Psi|, \quad \mathfrak{L}_\infty^d(P_\Omega, P_\Psi) = \max_{1 \leq i \leq q_1} |d_i^\Omega - d_i^\Psi|.$$

2.2 Chain complex and boundary operator

Boundary operator A p -chain is a sum of p -simplices denoted as $c' = \sum_i \alpha_i \sigma_i$, where σ_i are the p -simplices and α_i are either 0 or 1 [16]. The collection of p -chains forms a group and the sequence of these groups is called a *chain complex*. To relate chain groups, we denote a boundary operator $\partial_p : C_p \rightarrow C_{p-1}$, where C_p denotes the p -th chain group. For an oriented p -simplex σ_p with the ordered vertex set, the boundary operator is defined as

$$\partial_p \sigma_p := \sum_{i=0}^p (-1)^i [v_0, v_1, \dots, \hat{v}_i, \dots, v_p], \tag{1}$$

where $[v_0, v_1, \dots, \hat{v}_i, \dots, v_p]$ is a $(p-1)$ -simplex generated from $\sigma_p = [v_0, v_1, \dots, v_p]$ excluding \hat{v}_i . The boundary operator maps a simplex to its boundaries. Thus, $\partial_2 \sigma_2$ maps a triangle to its three edges. We can algebraically show that [16]

$$\partial_{p-1} \partial_p \sigma_p = 0.$$

The boundary map forms a sequence of chain groups as

$$C_2 \xrightarrow{\partial_2} C_1 \xrightarrow{\partial_1} C_0 \rightarrow 0.$$

For numerical implementation, we represent the boundary operator ∂_p using the boundary matrix \mathcal{B}_p [60,26]

$$(\mathcal{B}_p)_{ij} = \begin{cases} 1, & \text{if } \sigma_{p-1}^i \subset \sigma_p^j \text{ and } \sigma_{p-1}^i \sim \sigma_p^j \\ -1, & \text{if } \sigma_{p-1}^i \subset \sigma_p^j \text{ and } \sigma_{p-1}^i \not\sim \sigma_p^j \\ 0, & \text{if } \sigma_{p-1}^i \not\subset \sigma_p^j \end{cases} \quad (2)$$

where σ_{p-1}^i is the i -th $(p-1)$ -simplex and σ_p^j is the j -th p -simplex. Notations \sim and $\not\sim$ denote similar (positive) and dissimilar (negative) orientations respectively.

Coboundary operator The cochains are functions on the chain group. A cochain group C^p is a set of linear functions that maps an element in the chain group C_p to a real number

$$C^p = \{f : C_p \rightarrow \mathbb{R}\}.$$

C^p is called the p -th cochain group and its elements are p -cochains. A 0-cochain group $C^0 = \{\mathcal{V}^i : C_0 \rightarrow \mathbb{R}\}$ is written as

$$\mathcal{V}^i(v_j) = \begin{cases} 1, & \text{if } i = j, \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

Similarly a 1-cochain group $C^1 = \{\mathcal{F}^{[i,j]} : C_1 \rightarrow \mathbb{R}\}$ can be defined as

$$\mathcal{F}^{[i,j]}([k,l]) = \begin{cases} 1, & \text{if } [v_i, v_j] \text{ and } [v_k, v_l] \text{ are the same 1-simplex} \\ 0, & \text{otherwise.} \end{cases} \quad (4)$$

Similar to the boundary operator, we also have a coboundary operator that relates the cochain groups. The coboundary map $\delta_p : C^p \rightarrow C^{p+1}$ is a linear map that takes a p -cochain $f \in C^p$ to a $(p+1)$ -cochain $\delta_p f \in C^{p+1}$ defined by

$$(\delta_p f)(\sigma_{p+1}) := \sum_{i=0}^{p+1} (-1)^i f([v_0, v_1, \dots, \hat{v}_i, \dots, v_{p+1}]). \quad (5)$$

The equation (5) requires the evaluation of functions defined on vertices, edges, triangles and higher dimensional simplexes. When $p = 0$, we get

$$(\delta_0 f)(\sigma_1) = f([v_1]) - f([v_0]).$$

When $p = 1$, we have

$$(\delta_1 f)(\sigma_2) = f([v_1, v_2]) - f([v_0, v_2]) + f([v_0, v_1]).$$

To illustrate the construction of boundary and coboundary operators, consider a graph (1-skeleton) $G = (V, E)$ with vertex set V and edge set E :

$$V = [v_1, v_2, v_3, v_4, v_5], \quad E = [e_{12}, e_{13}, e_{23}, e_{24}, e_{35}, e_{45}]$$

with $e_{ij} = [v_i, v_j]$ in Figure 3. We can also have set of triangles T identified based on the edge connectivity as $t_{ijk} = [v_i, v_j, v_k] \in T$ if $[v_i, v_j], [v_i, v_k], [v_j, v_k] \in E$. The boundary matrix \mathcal{B}_1 is given by

$$\mathcal{B}_1 = \begin{pmatrix} e_{12} & e_{13} & e_{23} & e_{24} & e_{35} & e_{45} \\ v_1 & -1 & -1 & 0 & 0 & 0 \\ v_2 & 1 & 0 & -1 & -1 & 0 \\ v_3 & 0 & 1 & 1 & 0 & -1 & 0 \\ v_4 & 0 & 0 & 0 & 1 & 0 & -1 \\ v_5 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} \quad (6)$$

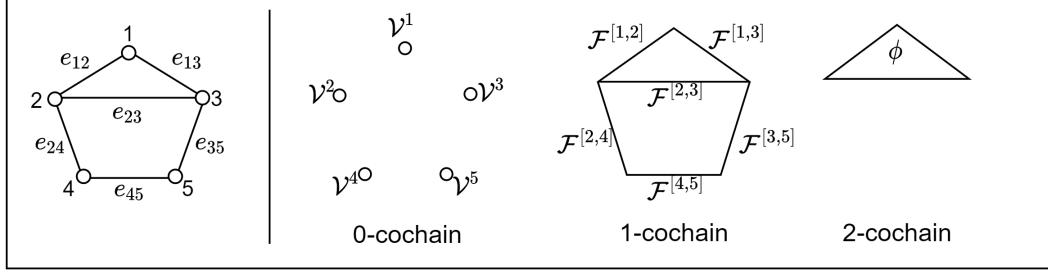


Fig. 3: Left: A typical network having 5 nodes and 6 edges and a triangle. The network representation using 0-cochain, 1-cochain and 2-cochain which contain functions defined on the vertices, edges and triangles respectively.

For coboundary operator δ_0 , we use equation (5). The function \mathcal{V}^i acting on the edge $e_{12} = [v_1, v_2]$ for instance, we get

$$(\delta_0 \mathcal{V}^1)([v_1, v_2]) = \mathcal{V}^1([\widehat{v}_1, v_2]) - \mathcal{V}^1([v_1, \widehat{v}_2]) = \mathcal{V}^1(v_2) - \mathcal{V}^1(v_1) = 0 - 1 = -1$$

Now we can compute \mathcal{V}^1 for all the edges in the edge set and show that

$$\delta_0(\mathcal{V}^1) = -\mathcal{F}^{[1,2]} - \mathcal{F}^{[1,3]}.$$

We can now compute $\delta_0(\mathcal{V}^i)$ for every i and build a coboundary matrix for δ_0

$$\begin{array}{c|ccccc} & \mathcal{V}^1 & \mathcal{V}^2 & \mathcal{V}^3 & \mathcal{V}^4 & \mathcal{V}^5 \\ \mathcal{F}^{[1,2]} & -1 & 1 & 0 & 0 & 0 \\ \mathcal{F}^{[1,3]} & -1 & 0 & 1 & 0 & 0 \\ \mathcal{F}^{[2,3]} & 0 & -1 & 1 & 0 & 0 \\ \mathcal{F}^{[2,4]} & 0 & -1 & 0 & 1 & 0 \\ \mathcal{F}^{[3,5]} & 0 & 0 & -1 & 0 & 1 \\ \mathcal{F}^{[4,5]} & 0 & 0 & 0 & -1 & 1 \end{array}. \quad (7)$$

It is seen that the coboundary matrix is simply the transpose of the boundary matrix \mathcal{B}_1 [13]. The boundary and coboundary operators when represented as matrices can be related as $\delta_p = \partial_{p+1}^T$ where T is the transpose. Similar to the boundary map, the coboundary map forms a sequence of cochain groups as

$$C^2 \xleftarrow{\delta_1} C^1 \xleftarrow{\delta_0} C^0 \leftarrow 0.$$

Similar to boundary operator satisfying $\delta_{p+1}\delta_p = 0$, the coboundary operator satisfies $\delta_p^T \delta_{p+1}^T = 0$.

The p -dimensional Hodge Laplacian $\mathcal{L}_p : C^p \rightarrow C^p$ is then written in terms of the boundary and coboundary operators as

$$\mathcal{L}_p = \delta_p^T \delta_p + \delta_{p-1} \delta_{p-1}^T. \quad (8)$$

For $p = 0$, $\Delta_0 = \delta_0^T \delta_0$ is the graph Laplacian. For $p = 1$, $\mathcal{L}_1 = \delta_1^T \delta_1 + \delta_0 \delta_0^T$ is the Helmholtz operator.

Functions on graphs We introduce two different functions defined on graphs, which we will use to compute the gradient and curl. The first function is the potential function, $s : V \rightarrow \mathbb{R}$ on the set of vertices that allows to assign a scalar value $s(v_i) = s_i$ to each vertex v_i . The second function is the function $X : E \rightarrow \mathbb{R}$ satisfying $X(v_i, v_j) = -X(v_j, v_i)$ if

$[v_i, v_j] \in E$ and $X(v_i, v_j) = 0$ otherwise. It can be seen that X is a skew symmetric matrix. If we let $X(v_i, v_j) = X_{ij}$, we need to have $X_{ij} = -X_{ji}$. The functions s and X are called 0- and 1-cochains respectively. 1-cochains is often referred as *edge flows* in the graph theory [28].

The gradient operator is written as

$$(\text{grad } s)(v_i, v_j) = s_j - s_i.$$

Similarly we can write the curl operator as

$$(\text{curl } X)(v_i, v_j, v_k) = X_{ij} + X_{jk} + X_{ki}.$$

The skew-symmetric nature of X ensures that curl X satisfies the conditions

$$\begin{aligned} (\text{curl } X)(v_i, v_j, v_k) &= (\text{curl } X)(v_j, v_k, v_i) = (\text{curl } X)(v_k, v_j, v_i) \\ &= -(\text{curl } X)(v_j, v_i, v_k) = -(\text{curl } X)(v_i, v_k, v_j) \\ &= -(\text{curl } X)(v_k, v_j, v_i). \end{aligned}$$

It is more convenient to represent the gradient and curl operators in terms of the coboundary operator δ using definition (5):

$$\begin{aligned} (\text{grad } s)(v_i, v_j) &= (\delta_0 s)(v_i, v_j) \\ (\text{curl } X)(v_i, v_j, v_k) &= (\delta_1 X)(v_i, v_j, v_k) \end{aligned} \tag{9}$$

The coboundary operators δ_0 and δ_1 act as the gradient and curl operators.

2.3 Hodge decomposition on graphs

The 1-cochain group (the space of edge flows) admits an orthogonal decomposition into subspaces which is expressed in terms of the coboundary operator [27]

$$C^1 = \text{im}(\delta_0) \oplus \text{im}(\delta_1^T) \oplus \text{ker}(\Delta_1). \tag{10}$$

The first term $\text{im}(\delta_0) = \text{im}(\text{grad}) = \delta_0 s$ denotes the subspace of edge flows that are gradient flows of a scalar potential function. The second term $\text{im}(\delta_1^T) = \delta_1^T \phi$ is the subspace of locally cyclic edge flows that have a non-zero curl along triangles. The third term $\text{ker}(\Delta_1)$ denotes a subspace of harmonic flows.

The Hodge decomposition splits edge flow X into three orthogonal components as

$$X = X_G + X_C + X_H,$$

where X_G , X_C and X_H are the gradient, curl and harmonic components respectively. Using the coboundary operator, the edge flow can be written as

$$X = \delta_0 s + \delta_1^T \phi + X_H. \tag{11}$$

In terms of the boundary operator, the Hodge decomposition can be written as

$$X = \partial_1^\top s + \partial_2 \phi + X_H. \tag{12}$$

The potential function s and the vector potential ϕ are defined on the nodes (0-simplices) and triangles (2-simplices) respectively. Since the components are orthogonal, each component can be evaluated *separately* as the projection of X onto the gradient, curl and harmonic subspaces respectively.

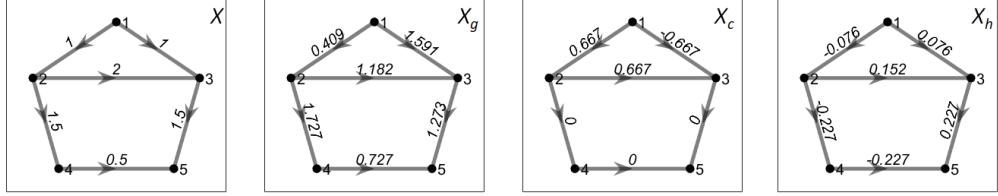


Fig.4: The Hodge decomposition of a network with 5 nodes and edge weights. The corresponding gradient flow X_G , curl flow X_C and harmonic flow X_H are shown from left to right.

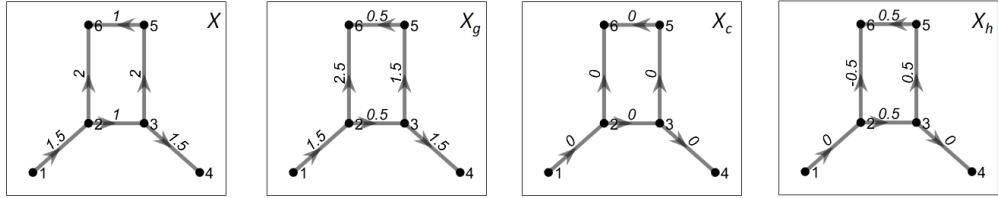


Fig.5: The Hodge decomposition with 1 loop. The corresponding gradient flow X_G curl flow X_C and harmonic flow X_H are shown with their magnitudes on the edges.

Let us work out the Hodge decomposition using the example in Figure 3 with edge flow

$$X = \begin{pmatrix} e_{12} \\ e_{13} \\ e_{23} \\ e_{24} \\ e_{35} \\ e_{45} \end{pmatrix} = \begin{pmatrix} 1.0 \\ 1.0 \\ 2.0 \\ 1.5 \\ 1.5 \\ 0.5 \end{pmatrix}$$

Let \mathcal{G} be a subspace of the space of edge flows C^1 . The orthogonal projection onto \mathcal{G} , denoted $P_{\mathcal{G}}$, is the linear operator that maps X to X_G . The gradient component is therefore obtained by minimizing the residual in the projection as:

$$P_{\mathcal{G}}X = \min_{s \in C^0} \|X - \delta_0 s\|.$$

The minimization is done by the least squares by solving

$$\begin{bmatrix} 1.0 \\ 1.0 \\ 2.0 \\ 1.5 \\ 1.5 \\ 0.5 \end{bmatrix} = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 1 \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \end{bmatrix}.$$

The least squares estimation (LSE) is given by

$$\hat{s} = [\hat{s}_1, \dots, \hat{s}_5]^\top = [-1.4000, -0.9909, 0.1909, 0.7364, 1.4636]^\top.$$

Subsequently, the gradient flow is estimated as $\hat{X}_G = \delta_0 \hat{s}$.

The projection of X onto the space of curl flow can be obtained similarly as

$$X_C = P_{\mathcal{C}}X = \min_{\phi \in C^1} \|X - \delta_1^T \phi\|,$$

where P_C , is the linear operator that maps X to X_C . Again we solve the problem by setting up a matrix equation

$$\begin{bmatrix} 1 \\ -1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \phi = \begin{bmatrix} 1.0 \\ 1.0 \\ 2.0 \\ 1.5 \\ 1.5 \\ 0.5 \end{bmatrix}.$$

The LSE of ϕ is given by $\hat{\phi} = 0.6667$. The curl flow is then estimated as $\hat{X}_C = \delta_1^T \hat{\phi}$.

The harmonic flow is estimated as the residual $\hat{X}_H = X - \hat{X}_G - \hat{X}_C$. Thus, the Hodge decomposition is given by

$$X_G = \begin{pmatrix} 0.409 \\ 1.591 \\ 1.182 \\ 1.727 \\ 1.273 \\ 0.727 \end{pmatrix} \quad X_C = \begin{pmatrix} 0.667 \\ -0.667 \\ 0.667 \\ 0.000 \\ 0.000 \\ 0.000 \end{pmatrix} \quad X_H = \begin{pmatrix} -0.076 \\ 0.076 \\ 0.152 \\ -0.227 \\ 0.227 \\ -0.227 \end{pmatrix},$$

which is displayed in Figure 4. The decomposition gives the gradient flow that sums to zero along cycles. For the triangle shaped cycle $e_{12} + e_{23} - e_{13}$ with the counter-clock wise orientation, we have $0.409 + 1.182 - 1.591 = 0$. Similarly, the the cycle with four nodes $e_{24} + e_{45} - e_{35} - e_{32}$, the gradient flow sum to $1.727 + 0.727 - 1.273 - 1.182 = 0$ within numerical accuracy.

We now consider another example shown in Figure 5 with exactly one loop (1-cycle). If we work out the algebraic details, we obtain the decompostion

$$X = \begin{pmatrix} e_{12} & 1.500 \\ e_{23} & 1.000 \\ e_{26} & 2.000 \\ e_{34} & 1.500 \\ e_{35} & 2.000 \\ e_{56} & 1.000 \end{pmatrix} \quad X_G = \begin{pmatrix} 1.500 \\ 0.500 \\ 2.500 \\ 1.500 \\ 1.500 \\ 0.500 \end{pmatrix} \quad X_C = \begin{pmatrix} 0.000 \\ 0.000 \\ 0.000 \\ 0.000 \\ 0.000 \\ 0.000 \end{pmatrix} \quad X_H = \begin{pmatrix} 0.000 \\ 0.500 \\ -0.500 \\ 0.000 \\ 0.500 \\ 0.500 \end{pmatrix}.$$

Loop and non-loop ratios The magnitude of an edge flow X can be measured by its l_2 -norm $\|X\|$. Since the gradient, curl and harmonic flows are the orthogonal, we have

$$\|X\|^2 = \|X_G\|^2 + \|X_C\|^2 + \|X_H\|^2 = \|X_G\|^2 + \|X_L\|^2.$$

We can then define the relative strength of each of the component as

$$\eta_G = \frac{\|X_G\|^2}{\|X\|^2}, \quad \eta_C = \frac{\|X_C\|^2}{\|X\|^2}, \quad \eta_H = \frac{\|X_H\|^2}{\|X\|^2}, \quad \eta_L = \frac{\|X_L\|^2}{\|X\|^2}. \quad (13)$$

Here, η_G , η_C and η_H measure the relative strengths of each component and satisfies

$$\eta_G + \eta_C + \eta_H = 1.$$

The ratio η_L is called the *loop ratio*, which quantifies the strength of the loop flow [35]. In contrast, we refer η_G as the non-loop ratio and we have

$$\eta_G + \eta_L = 1.$$

Figure 6 shows the variation of the non-loop and loop ratios during graph filtration on a graph with 5 nodes and 10 edges. Each edge has a unique edge weight and the edge set is given as $E = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0]^\top$. The graph filtration is performed

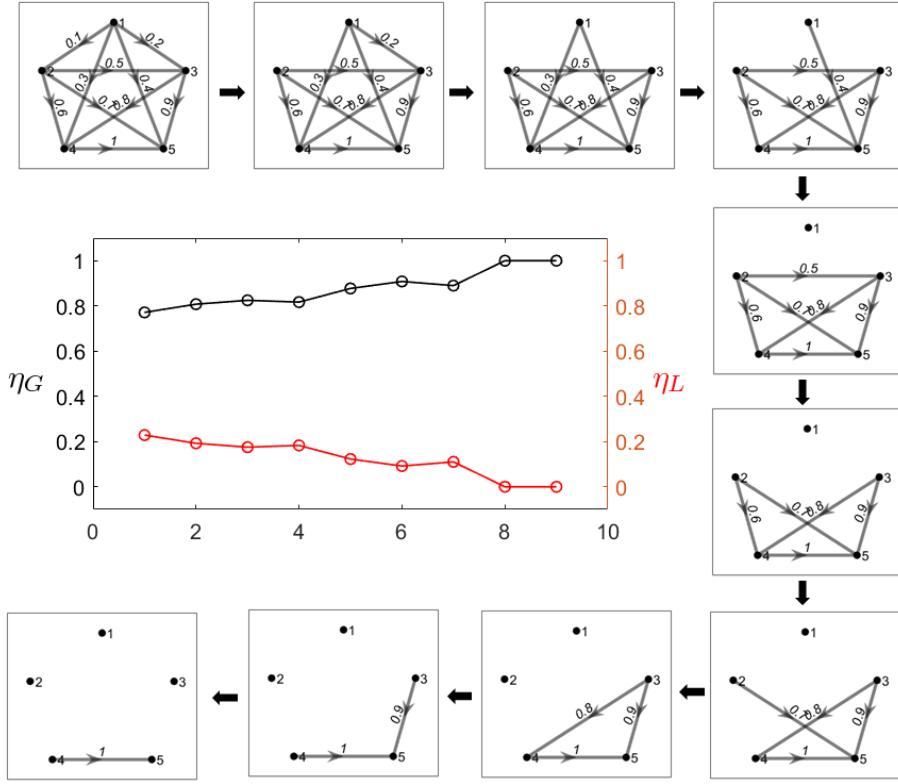


Fig. 6: The variation of the non-loop ratio (black line) and loop ratio (red line) during graph filtration. The sum of the non-loop ratio and loop ratio at any filtration step is 1.0. Since the loops are destroyed during filtration the loop ratio eventually becomes zero.

by sorting the edges in ascending order and removing one edge at a time sequentially until we complete all the edges in the edge set. We then compute the gradient ratio and loop ratio at each filtration step and track their variation until all the edges are removed. The sum of the non-loop ratio and loop ratio at any filtration step is 1.0. Since the loops are destroyed during filtration the loop ratio eventually becomes zero.

Diffusion behavior of Hodge decomposition We investigate the diffusion behavior of the Hodge decomposition. We first study the diffusion dynamics of the edge flow X and then analyze the components of the Hodge decomposition separately. Consider diffusion equation [13,30]

$$\frac{dX(t)}{dt} = -\mathcal{L}_1 X(t) \quad (14)$$

with initial condition $X(t=0) = X_0$. The solution is given in the matrix form as

$$X(t) = e^{-\mathcal{L}_1 t} X_0, \quad (15)$$

where $e^{-\mathcal{L}_1 t}$ is a matrix-valued function of t known as the Laplacian exponential diffusion kernel [61]. Let $\mathcal{L}_1 = \mathbf{U} \Lambda \mathbf{U}^T$ be the spectral decomposition, where Λ is the diagonal matrix of eigenvalues of \mathcal{L}_1 with corresponding eigenvectors in the columns of the matrix \mathbf{U} . The Laplacian exponential can be written as [62]

$$e^{-\mathcal{L}_1 t} = \mathbf{U} e^{-\Lambda t} \mathbf{U}^T. \quad (16)$$

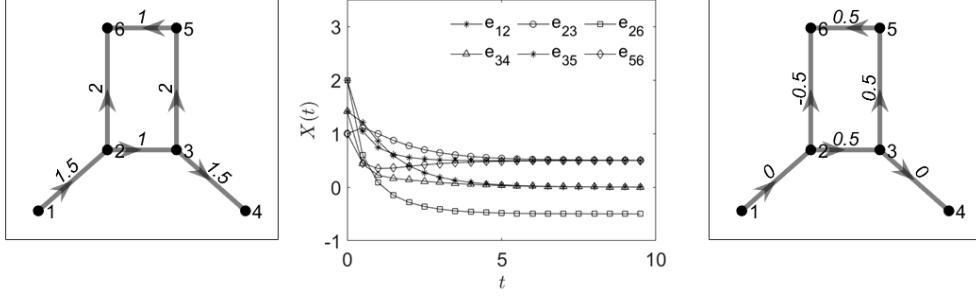


Fig. 7: Left: A network with 6 nodes and 6 edges with a specified edge flow. Middle: The diffusion dynamics of the edge flow showing the edge weights converging to a steady state. It is noticed that except the edges that contribute to the cycle the weight values on the other edges tends to zero.

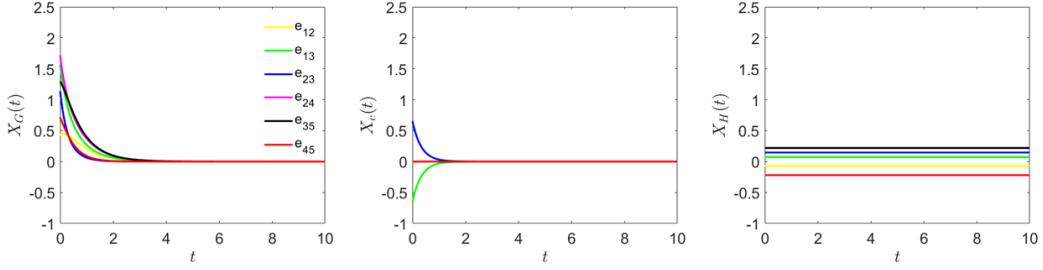


Fig. 8: The diffusion behavior of gradient, curl and harmonic flows for the network for Figure 5 network.

We can then rewrite (15) as

$$X(t) = \mathbf{U} e^{-\Lambda t} \mathbf{U}^T X_0. \quad (17)$$

Let λ_j be the j -th eigenvalue with corresponding eigenvector \mathbf{U}_j . The matrix exponential $e^{-\Lambda t}$ can be written as

$$e^{-\Lambda t} = \begin{bmatrix} e^{\lambda_1 t} & & & & \\ & e^{\lambda_2 t} & & & \\ & & \ddots & & \\ & & & & e^{\lambda_{|E|} t} \end{bmatrix}.$$

Subsequently, we have

$$X(t) = \sum_{j=1}^{|E|} e^{-\lambda_j t} (\mathbf{U}_j^T X_0) \mathbf{U}_j. \quad (18)$$

The equation (18) is used to perform diffusion over 1-simplices (edges).

Consider Figure 5 example, where we have the initial edge flow represented as a column vector $X_0 = [1.5, 1.0, 2.0, 1.5, 2.0, 1.0]^\top$, which is taken as the initial condition at $t = 0$. We use the equation (15) and perform the diffusion up to time $t = 10$ in incremental steps of $dt = 0.01$. Figure 7 shows the diffusion dynamics of each edge in the edge flow. As the time increases, the edge flow reaches to a steady state, which lies in the harmonic subspace. At the steady state, we have $\frac{dX(t)}{dt} = 0$ and the equation (14) is reduced to $\mathcal{L}_1 X(t) = 0$. The solution lies in the kernel space $\ker(\mathcal{L}_1)$, which is the harmonic subspace. Thus, the harmonic component is simply the steady state solution.

In order to study the diffusion behavior of the gradient, curl and harmonic flows, we use equation (14). Substituting $X(t) = X_G(t) + X_C(t) + X_H(t)$, we obtain

$$\frac{d[X_G(t) + X_C(t) + X_H(t)]}{dt} = -\mathcal{L}_1 X_G(t) - \mathcal{L}_1 X_C(t) - \mathcal{L}_1 X_H(t) \quad (19)$$

The Hodge-Laplacian in terms of the coboundary operator is written as $\mathcal{L}_1 = \delta_1^T \delta_1 + \delta_0 \delta_0^T$. The gradient and curl components are expressed as $X_G = \delta_0 s$ and $X_C = \delta_1^T \phi$. Then, equation (19) is written as

$$\begin{aligned} \frac{d[X_G(t) + X_C(t) + X_H(t)]}{dt} &= -\delta_1^T \delta_1 \delta_0 s - \delta_0 \delta_0^T \delta_0 s \\ &\quad - \delta_1^T \delta_1 \delta_1^T \phi - \delta_0 \delta_0^T \delta_1^T \phi \\ &\quad - \mathcal{L}_1 X_H(t). \end{aligned}$$

Using the coboundary operator properties

$$\delta_1 \delta_0 = 0, \quad \delta_0^T \delta_1^T = 0, \quad \mathcal{L}_1 X_H = 0,$$

we obtain

$$\frac{d[X_G(t) + X_C(t) + X_H(t)]}{dt} = -\delta_0 \delta_0^T \delta_0 s - \delta_1^T \delta_1 \delta_1^T \phi. \quad (20)$$

Since these subspaces are orthogonal, we can decouple them into three linear set of differential equations:

$$\frac{d[X_G(t)]}{dt} = -\delta_0 \delta_0^T X_G, \quad \frac{d[X_C(t)]}{dt} = -\delta_1^T \delta_1 X_C, \quad \frac{d[X_H(t)]}{dt} = 0. \quad (21)$$

The solutions to equations (21) are given as

$$X_G(t) = e^{-\delta_0 \delta_0^T t} X_G(0), \quad X_C(t) = e^{-\delta_1^T \delta_1 t} X_C(0), \quad X_H(t) = X_H(0). \quad (22)$$

Figure 8 shows the diffusion behavior of gradient, curl and harmonic flows for Figure 5 network. The gradient and curl flows exponentially decrease to zero exponentially while the harmonic component remains constant.

2.4 Modeling cycles using the Hodge decomposition

We propose a method to identify and extract independent 1-cycles from a network using the curl and harmonic components of the Hodge decomposition. The cycles are often enumerated and computed using the brute-force depth-first search algorithms in the graph theory [63]. The cycles can be obtained as a linear combination of eigenvectors corresponding to zero eigenvalues of the Hodge Laplacians [64,13]. In persistent homology, the cycles are modeled manipulating a matrix associated with the boundary operator [16,65]. However, such methods do not provide clear identifiable cycles as individual features. In this paper, we present a new approach using the Hodge decomposition to extract the 1-cycle explicitly.

Typically, a complete graph contains many cycles and it is time consuming to obtain all of them. However, this task is greatly simplified if we decompose the graph into several subgraphs containing only one cycle (Figure 9). For a given graph, we perform the birth-death decomposition [44]. The birth set is the maximum spanning tree (MST) while the remaining edges are the death set. MST itself does not contain any cycles. Each time an edge from the death set is added to MST, a cycle is born. The process is repeated sequentially till we exhaust all the edges in the death set. Since each subgraph differs at least by an edge, we have the complete set of algebraically independent cycles which can be used as a basis [13].

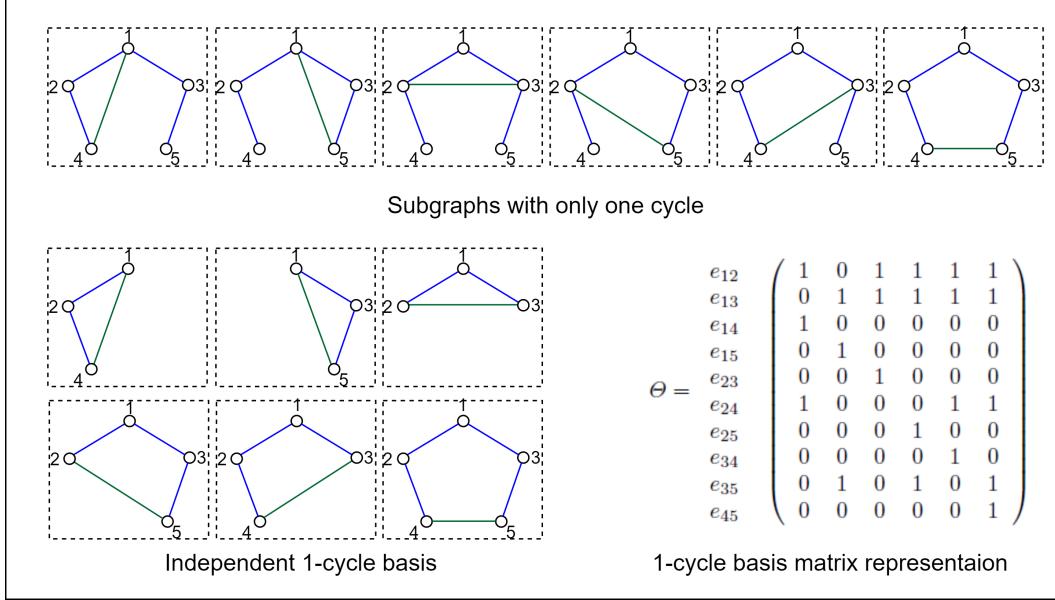


Fig. 9: Top: The formation of subgraphs containing only one cycle using the MST and one edge from the death set. Bottom left: The independent 1-cycle basis obtained using the curl and harmonic component of the Hodge decomposition performed on the subgraphs. The cycles with 3 edges are identified using the curl component while the cycles with more than 3 edges are identified using the harmonic component. Bottom right: The matrix representation of the 1-cycle basis with each column corresponding to an algebraically independent cycle. The basis is constructed by assigning the value 1 to the edges that constitute the cycle and all other edges are set to zero.

Although the subgraphs have only one cycle they also contain edges that are not a part of the cycle. We eliminate these edges that do not constitute the cycle using the Hodge decomposition. The harmonic component of the Hodge decomposition identifies itself with the 1-cycles and the curl component captures the triangles. We exploit this result to accurately locate and extract the 1-cycle. The algebraically independent 1-cycles with more than three edges can be systematically extracted from a network by considering the non-zero entries of the harmonic component while the 1-cycles with three edges (triangles) are associated with the curl flow. The 1-cycle basis is constructed by assigning the value 1 to the edges that constitute the cycle and all other edges are set to zero. We then construct a matrix representation of the 1-cycle basis with each column corresponding to an algebraically independent cycle. Although these columns in the basis matrix are independent of each other they are not orthonormal. In Figure 9, we have $q_1 = (p - 1)(p - 2)/2 = 6$ independent 1-cycles formed from the complete graph with $p = 5$ nodes and $q = p(p - 1)/2 = 10$ edges. The basis set Θ is formed by the collection of independent 1-cycle based basis vectors as shown in Figure 9 on the right. Once we have the 1-cycle basis, any cycle in the graph can be represented as a linear combination of the cycle basis.

2.5 Statistical Inference on the Hodge decomposition

Given a collection of Hodge decompositions, we develop a new statistical inference procedure for comparing Hodge decompositions topologically. Consider the Hodge decomposition from the complete graph in Figure 10. We perform the birth-death decomposition on the non-loop and loop flows separately. Since the birth and death sets are different, each decomposition yield topologically distinct pattern. Therefore it is worthwhile to examine the components

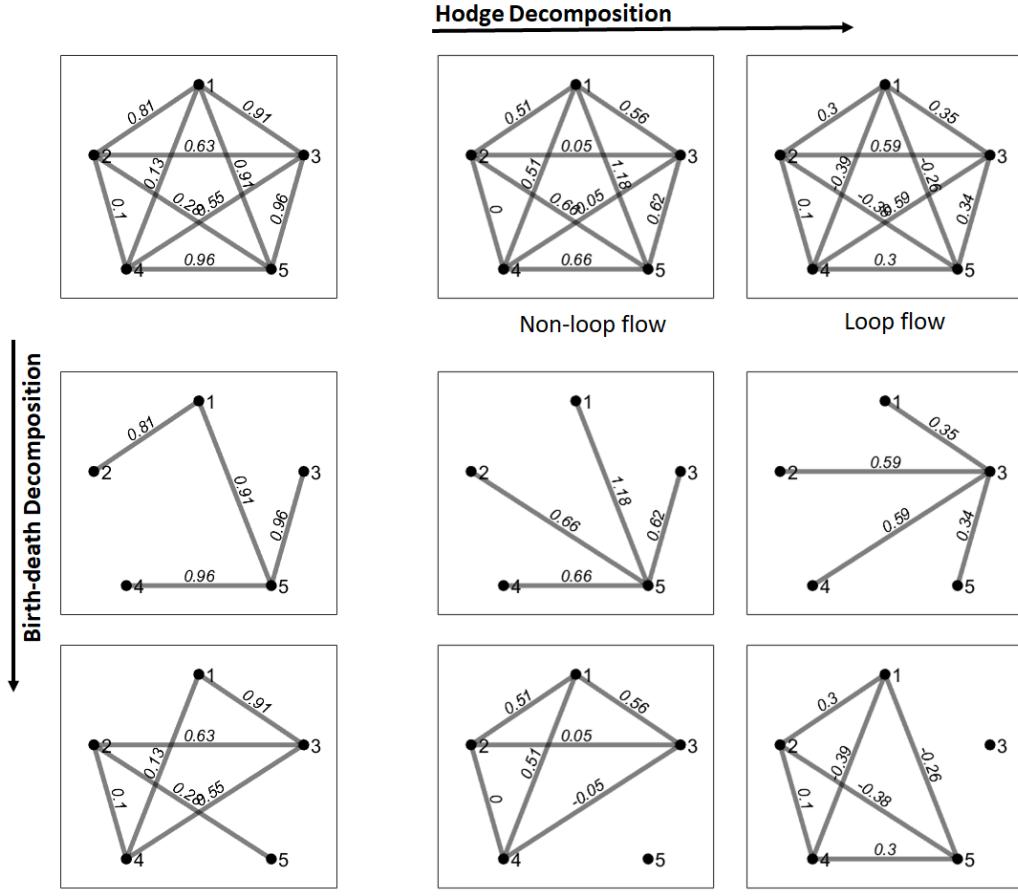


Fig. 10: Top: The Hodge decomposition of an edge flow into non-loop flow and loop flow. The middle and last columns are the birth-death decomposition of the edge flow, non-loop flow and loop flow respectively. Clearly we see that the birth and death edge sets of the non-loop and loop flows are very different from the birth set and death set of the original edge flow.

independently for their topological characteristics. To this end we propose statistical inference procedures using the components of Hodge decomposition. We will develop statistical inference procedure based on Wasserstein distance using the non-loop (gradient) and loop flows (curl component). We will also develop the statistical inference procedure based on common 1-cycles computed using the curl and harmonic components.

Inference using the ∞ -Wasserstein distance We propose a statistical inference procedure based on the Wasserstein distance applied to the birth and death values. We first outline the procedure for the loop component and later use the same procedure for the non-loop component. Let $\Omega = \{\Omega_1, \Omega_2, \dots, \Omega_m\}$ and $\Psi = \{\Psi_1, \Psi_2, \dots, \Psi_n\}$ represent a collection of m and n networks respectively. Let each of these networks be subjected to the Hodge decomposition to obtain their respective loop and non-loops flows. We first analyze the loop component from the groups Ω and ψ and examine their topological equivalence. We are interested in testing whether the topology of Hodge decomposition is equivalent between the groups. We propose a test statistic based on the decomposed edge set as follows. Let

$$b_1^{\Omega_i} \leq b_2^{\Omega_i} \leq \dots \leq b_{q_0}^{\Omega_i}$$

be the sorted birth values in Ω_i . q_0 is the number of edges in the birth set. The average birth value across subjects is given by

$$\bar{b}_1^\Omega \leq \bar{b}_2^\Omega \leq \cdots \leq \bar{b}_{q_0}^\Omega,$$

where

$$\bar{b}_j^\Omega = \frac{b_j^{\Omega_1} + b_j^{\Omega_2} + \cdots + b_j^{\Omega_m}}{m} \quad (23)$$

is the average birth value of j -th birth edge in group Ω . Similarly we sort birth values in Ψ_i as

$$b_1^{\Psi_i} \leq b_2^{\Psi_i} \leq \cdots \leq b_{q_0}^{\Psi_i}$$

and obtain the average birth values across subjects

$$\bar{b}_1^\Psi \leq \bar{b}_2^\Psi \leq \cdots \leq \bar{b}_{q_0}^\Psi,$$

Subsequently, we use the maximum difference

$$\mathcal{L}_\infty^b(\Omega, \Psi) = \max_{1 \leq j \leq q_0} |\bar{b}_j^\Omega - \bar{b}_j^\Psi| \quad (24)$$

to measure the topological difference. The topological distance $\mathcal{L}_\infty^b(\Omega, \Psi)$ is the ∞ -Wasserstein distance of persistent diagrams on connected components. We then extend the same idea to the death values and write the distance as

$$\mathcal{L}_\infty^d(\Omega, \Psi) = \max_{1 \leq j \leq q_1} |\bar{d}_j^\Omega - \bar{d}_j^\Psi|,$$

where q_1 is the number of edges in the death set. $\mathcal{L}_\infty^d(\Omega, \Psi)$ is the ∞ -Wasserstein distance of persistent diagrams on 1-cycles.

We now define our test statistic as the sum of \mathcal{L}_∞^b and \mathcal{L}_∞^d as

$$\mathcal{L}_\infty(\Omega, \Psi) = \max_{1 \leq j \leq q_0} |\bar{b}_j^\Omega - \bar{b}_j^\Psi| + \max_{1 \leq j \leq q_1} |\bar{d}_j^\Omega - \bar{d}_j^\Psi| \quad (25)$$

in discriminating between the two groups of networks. The test statistic has two terms corresponding to the ∞ -Wasserstein distances computed on the average birth and death sets respectively. The small values of \mathcal{L}_∞ indicates topological similarity while large values corresponds to topological dissimilarity. The combined distances are expected to be more robust compared to using the birth and death sets independently. Using the proposed test statistic $\mathcal{L}_\infty(\Omega, \Psi)$, we test the topological equivalence of the two groups Ω and Ψ . Under the null hypothesis that the two groups are topologically equivalent, $\mathcal{L}_\infty(\Omega, \Psi)$ should be close to zero, and its deviation from zero indicates topological difference. The null distribution of $\mathcal{L}_\infty(\Omega, \Psi)$ is not known and will be estimated using the permutation test and the p -value computed. There is no multiple comparisons issue.

Statistical inference on common 1-cycles across networks We propose a statistical inference procedure for discriminating two groups using the 1-cycle basis. At the group level, we identify the group-level fixed cycle basis that is common across subject by using the network template $\bar{\mathcal{S}}$, which is computed as the average of all the individual connectivity matrices. The algebraically independent cycles are then extracted from the network template to build the common cycle basis. To reflect the individual network variability, the subject-level network connectivity \mathcal{W} is expanded in terms of the cycle basis as

$$\mathcal{W} = \sum_{j=1}^{q_1} \alpha_j \Theta_j. \quad (26)$$

The coefficients $\boldsymbol{\alpha} = [\alpha_1, \dots, \alpha_{q_1}]^\top$ are estimated in the least squares fashion as

$$\hat{\boldsymbol{\alpha}} = (\boldsymbol{\Theta}^\top \boldsymbol{\Theta})^{-1} \boldsymbol{\Theta}^\top \mathcal{W},$$

where $\boldsymbol{\Theta}$ is a matrix whose columns $\boldsymbol{\Theta}_j$ represent the 1-cycle basis of the network template $\bar{\mathcal{S}}$. Let $\bar{\alpha}_j^1$ and $\bar{\alpha}_j^2$ be the mean of the j -th cycle of network in group Ω and Ψ respectively. Using the cycle basis expansion coefficients, we propose the following statistic for testing whether there is a statistically significant topological difference between the two groups:

$$\mathcal{T}(\Omega, \Psi) = \max_{1 \leq j \leq q_1} |\bar{\alpha}_j^1 - \bar{\alpha}_j^2| \quad (27)$$

The statistical significance is determined using the permutation test [21].

3 Validations

We conducted extensive validation studies with random network simulations with the ground truth [66, 67, 45, 68, 69, 41]. We assume a network to be *strongly connected* if most of the edge weights have higher connectivity values. To obtain strongly connected random networks, we generated complete graphs whose edge values are sampled from a skewed distribution. In our simulations, the connectivity matrices are formed from complete graphs whose edge weights are drawn from a Beta distribution [45]. The Beta distribution $Beta(\alpha^*, \beta^*)$ is a family of continuous probability distributions defined on the interval $[0, 1]$ and parametrized by two positive shape parameters α^* and β^* that appear as exponents of the random variable and control the shape of the distribution [70]. This allows us to generate networks of varying strength of connectivities and compare their topological similarity and dissimilarity. We then construct networks of desired connectivity strengths by drawing edge weights from these distributions. Since the probability of drawing a zero edge weight is nil we always have a complete graph.

Once we generate the random network, we perform the Hodge decomposition to obtain the non-loop (gradient) and loop (curl) components. These components are assessed individually to quantify their topological characteristics. We treat the gradient component as non-loop flow and the sum of curl and harmonic flow together as loop flow. The harmonic component is zero for a complete graph and hence the loop component is simply the curl. The statistical inferences are performed using the permutation test with 100000 permutations. The simulations are independently repeated for 10 times and the average p-values are reported.

3.1 Simulation study using Beta distributions

We generated random networks using the Beta distribution with parameters $(2, 2)$, $(2, 4)$ and $(4, 2)$. The shape of distributions corresponding to these parameters and their corresponding connectivity matrices are shown in Figure 11. The connectivity matrix following $Beta(4, 2)$ has more stronger connections (higher edge weights) compared to the connectivity matrix following $Beta(2, 4)$. In our simulation studies, we set the number of nodes to be $p = 20$ which sets the number of edges to be $q = p(p - 1)/2 = 190$ for each simulated network. We generated 5, 10, 50 and 100 number of networks of same type and compared across them.

To evaluate the topological similarity and dissimilarity of networks, we used the test statistic (25). The test is performed on the non-loop component (gradient network) and loop component (curl network). Table 1 display the resulting p-values of testing network differences on the non-loop and loop components. The p-values are small for networks with differences while they are large when the network has no difference. However, in comparing 5 networks vs. 5 networks, the test is not necessarily performing well due to the small sample size. *The study demonstrates the topological characteristics of the original network is also present in the Hodge decompositions and can be quantified using the Wasserstein distance.*

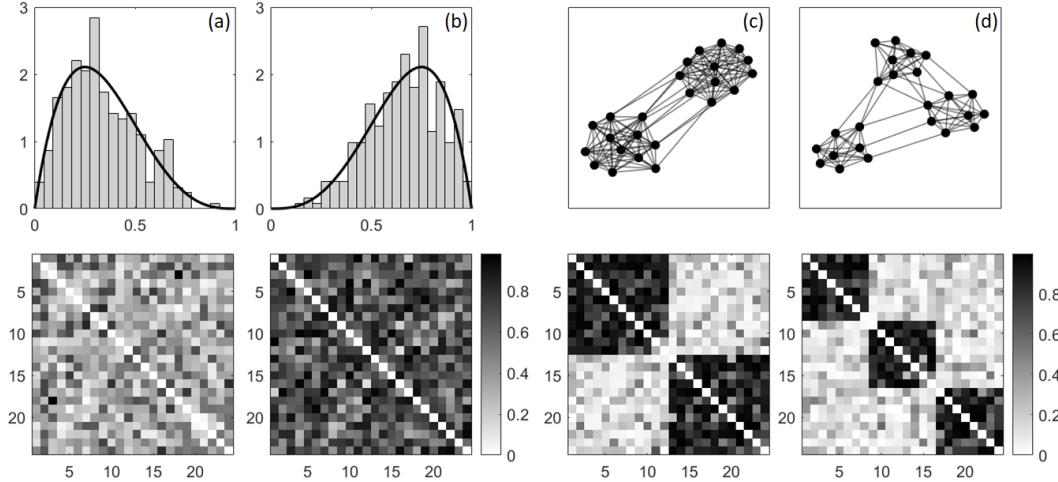


Fig. 11: Edge weights following the Beta distributions with parameters (a) ($\alpha^* = 2, \beta^* = 4$) (b) ($\alpha^* = 4, \beta^* = 2$) with their corresponding connectivity matrices. The plot of random modular graphs obtained using Beta distributions with (c) two and (d) three modules. The networks are thresholded at 0.4 to enable better display of the modules.

Table 1: The performance results of the ∞ -Wasserstein distance on the loop (first numbers) and non-loop (second numbers) components summarized as the average p-values for testing various combinations of beta distributions. Smaller p-values are better when there are network differences (top rows) and larger p-values are better when there are no network differences (bottom rows).

Beta distributions	10 networks	50 networks	100 networks
(2,2) vs. (2,4)	0.0000, 0.0002,	0.0000, 0.0000	0.0000, 0.0000
(2,2) vs. (4,2)	0.0000, 0.0000	0.0000, 0.0000	0.0000, 0.0000
(2,4) vs. (4,2)	0.0000, 0.0000	0.0000, 0.0000	0.0000, 0.0000
(2,2) vs. (2,2)	0.8867, 0.8084	0.6255, 0.6324	0.1876, 0.7303
(2,4) vs. (2,4)	0.9110, 0.4001	0.2103, 0.1276	0.6592, 0.4227
(4,2) vs. (4,2)	0.7405, 0.1932	0.4774, 0.8713	0.3382, 0.4423

3.2 Simulation study using random modular networks

To test the performance of the method further, we used random modular networks. A typical random modular network ensures that nodes within a module are strongly connected whereas the nodes from different modules have weak connections. In order to construct such a network, we assign positive integer values for parameters α^* and β^* . We then choose the edge weights such that the edges within the same module are assigned values from $Beta(\alpha^*, \beta^*)$ with $\alpha^* > \beta^*$ while the edges from different modules are assigned values from $Beta(\beta^*, \alpha^*)$. We used $\alpha^* = 5, \beta^* = 1$. Figure 11 displays random modular graphs obtained in this fashion. The networks are thresholded at 0.4 to enable better display of the modules. We tested if the topological features in the non-loop flow and loop flows can be used in discriminating networks having different modules. We simulated networks with $p = 24$ number of nodes and $c = 2, 3, 6$ number of modules, where nodes are distributed evenly among the modules. We generated four different types of random modular networks (Type I, II, III, IV) by fixing $\alpha^* = 5$ and varying β^* from 1 to 4 (Figure 12). Lower values of β^* makes the modular structure more pronounced. For each parameter, we generated three random networks having 2, 3 and 6 modules. Figure 13 shows the Hodge decomposition of

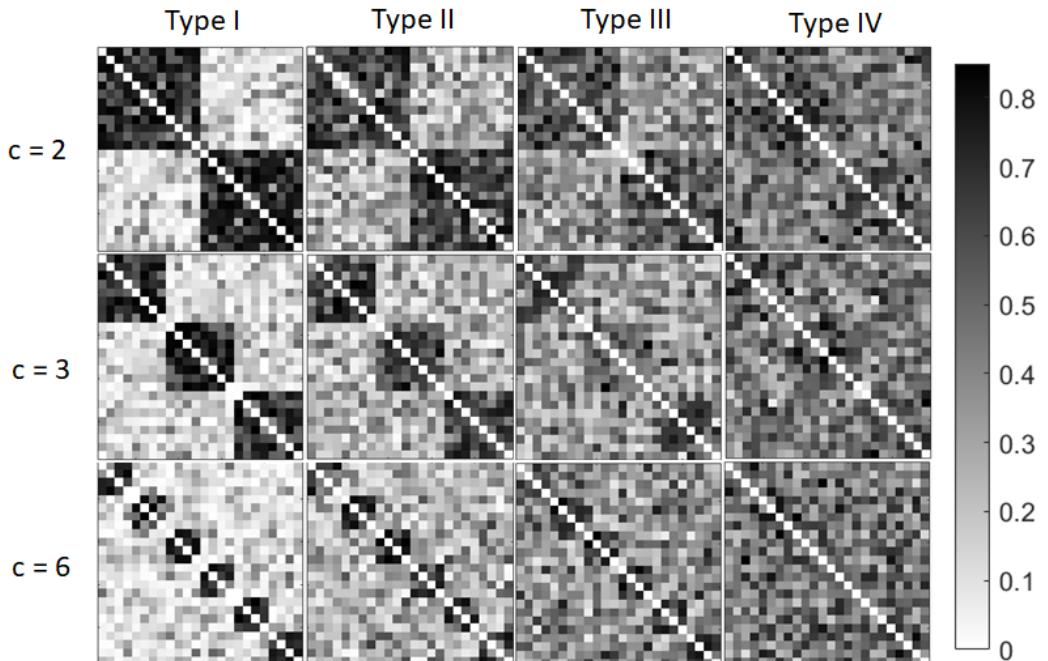


Fig. 12: Random modular networks obtained using Beta distributions with $\alpha^* = 0.5$ and $\beta^* = 1, 2, 3, 4$ corresponding to type I, II, III and IV networks respectively. For each type, we have $c = 2, 3, 6$ modules respectively.

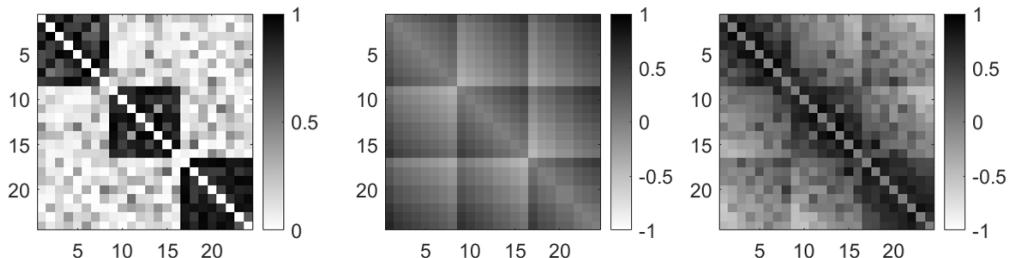


Fig. 13: A random network with three modules (left) and its corresponding non-loop flow (middle) and loop flow (right). Each Hodge components carries the underlying modularity information.

a network with three modules. The non loop and loop components preserve modularity but they exhibit distinctly different connectivity patterns.

The simulation studies were then performed on the modular networks with specific number of modules to examine if the non-loop and the loop components can preserve the topological dissimilarity. Tables 3 and 3 show the performance results for non-loop (gradient) and loop (curl) components. When there are too much noise and the strength of modularity is low (type-IV), it gets difficult to distinguish networks with the proposed method. However, in most cases, each components were able to discriminate networks when there are differences. The MATLAB codes used for this particular simulation study are provided in <https://github.com/laplcebeltrami/hodge>. The script `SIMULATION_hoggedecompose.m` was used for generating the performance tables.

Table 2: The performance results of the ∞ -Wasserstein distance on the non-loop component with different number of nodes p and modules c . Smaller p-values are better when there are network differences (top rows) and larger p-values are better when there are no network differences (bottom rows).

Nodes p	Modules c	Type of Network			
		Type-I	Type-II	Type-III	Type-IV
12 vs. 12	2 vs. 3	0.0000	0.0005	0.0001	0.1318
	3 vs. 6	0.0000	0.0007	0.0001	0.0792
	2 vs. 3	0.0000	0.0001	0.0000	0.0467
	3 vs. 6	0.0000	0.0001	0.0000	0.0250
18 vs. 18	2 vs. 3	0.0000	0.0001	0.0000	0.0467
	3 vs. 6	0.0000	0.0001	0.0000	0.0250
24 vs. 24	2 vs. 3	0.0000	0.0003	0.0008	0.0035
	3 vs. 6	0.0000	0.0000	0.0001	0.0687
	6 vs. 6	0.7228	0.4717	0.4324	0.4850
24 vs. 24	3 vs. 3	0.1067	0.6825	0.7070	0.4471
	6 vs. 6	0.8212	0.5070	0.3743	0.5694

Table 3: The performance results of the ∞ -Wasserstein distance on the loop component with different number of nodes p and modules c . Smaller p-values are better when there are network differences (top rows) and larger p-values are better when there are no network differences (bottom rows).

Nodes p	Modules c	Type of Network			
		Type-I	Type-II	Type-III	Type-IV
12 vs. 12	2 vs. 3	0.0000	0.0004	0.0018	0.0152
	3 vs. 6	0.0000	0.0000	0.0020	0.0160
	2 vs. 3	0.0000	0.0000	0.0017	0.0049
	3 vs. 6	0.0000	0.0001	0.0002	0.0043
18 vs. 18	2 vs. 3	0.0000	0.0000	0.0001	0.1646
	3 vs. 6	0.0000	0.0000	0.0003	0.0526
24 vs. 24	2 vs. 3	0.6329	0.9021	0.7862	0.7950
	3 vs. 3	0.5029	0.7347	0.7363	0.7329
	6 vs. 6	0.6277	0.8891	0.7642	0.9007

3.3 Simulation study using the common cycle basis

Using the same simulation model and setting in Section 3.2, we determined the performance of the common cycle basis method presented in Section 2.5. The random modular networks typically shows clusters that contain many small cycles within the clusters and the clusters themselves are in turn connected by larger cycles. Therefore it would be interesting to examine the common cycles that constitute the modular networks and further use them as topological features to discriminate networks with different modules. Table 4 displays the performance results. The common cycle basis based method generally performs well except for the random modular network of Type IV, which has too much noise and thus difficult to discriminate the networks using topological features.

3.4 Comparison against other topological distances

Using the same simulation model and setting in Section 3.2 (type-I only), we compared the proposed method against often used other topological distances. The topological similarity or dissimilarity can be quantified using the distance measures or losses such as bottleneck [71], Gromov-Hausdorff (GH) [41,42]. In brain network analysis, these distances are modified

Table 4: The performance results of the ∞ -Wasserstein distance using common 1-cycle basis summarized as average p-values. Smaller p-values are better when there are network differences (top rows) and larger p-values are better when there are no network differences (bottom rows).

Nodes	Modules	Type of Network			
		Type-I	Type-II	Type-III	Type-IV
p	c				
12 vs. 12	2 vs. 3	0.0000	0.0000	0.0137	0.1639
	3 vs. 6	0.0000	0.0000	0.0007	0.6223
18 vs. 18	2 vs. 3	0.0000	0.0006	0.0187	0.5039
	3 vs. 6	0.0000	0.0000	0.0117	0.2157
24 vs. 24	2 vs. 3	0.0000	0.0000	0.0182	0.0008
	3 vs. 6	0.0000	0.0001	0.0123	0.0883
24 vs. 24	2 vs. 2	0.2396	0.4480	0.4835	0.7806
	3 vs. 3	0.6475	0.2520	0.7714	0.3365
	6 vs. 6	0.5906	0.8201	0.3472	0.3467

Table 5: The performance results of the bottleneck and Gromov-Hausdorff (GH) distances compared against the Wasserstein distance on the non-loop and loop components. Smaller p-values are better when there are network differences (top rows) and larger p-values are better when there are no network differences (bottom rows).

Nodes	Modules	Graph distance measures			
		Bottleneck	GH	nonloop	loop
p	c				
12 vs. 12	2 vs. 3	0.7056	0.7786	0.0000	0.0081
	3 vs. 6	0.5589	0.8816	0.0077	0.0000
18 vs. 18	2 vs. 3	0.3488	0.7822	0.0087	0.0078
	3 vs. 6	0.5596	0.7269	0.0000	0.0000
24 vs. 24	2 vs. 3	0.1519	0.1575	0.0075	0.0080
	3 vs. 6	0.8153	0.2625	0.0085	0.0000
24 vs. 24	2 vs. 2	0.8144	0.6885	0.7322	0.3500
	3 vs. 3	0.9397	0.2398	0.9923	0.2228
	6 vs. 6	0.1337	0.3578	0.9906	0.7718

to accommodate topological features generated from persistent homology such as barcodes, persistent diagrams and graph filtrations [23]. For the permutation test, the average network of each group in each permutation is used as input to compute the bottleneck and GH distances. The average p-values for 10 independent simulations are reported in Table 5. From the table, we observe that our ∞ -Wasserstein distance based test statistic performs well on both the loop and non-loop flows.

The ∞ -Wasserstein distance for k -cycles is the bottleneck distance. However, the ∞ -Wasserstein distances \mathcal{L}_∞ combines the ∞ -Wasserstein distances for 0-cycles and 1-cycles. The bottleneck and GH distances did not perform well for dissimilar networks whereas \mathcal{L}_∞ were able to discriminate networks in the non-loop (gradient) and loop (curl) components (top rows) when there is network differences. In the case of no network differences (bottom rows), all the distances perform well. Therefore the modularity in the network is well identified by the non-loop and loop components of the Hodge decomposition and our ∞ -Wasserstein distance can discriminate the modularity differences.

4 Application

4.1 Brain imaging data and preprocessing

The 412 subjects brain imaging data used in this study was obtained from the Human Connectome Project (HCP) [72,73]. The resting-state functional magnetic resonance images (rs-fMRI) were collected on Siemens 3T Connectome Skyra scanners using the gradient-echoplanar imaging (EPI) sequence with multiband factor 8, repetition time (TR) $720ms$, time echo (TE) $33.1ms$, flip angle 52° , 104×90 (RO \times PE) matrix size, 72 slices, $2mm$ isotropic voxels. The scans were acquired for approximately 15 minutes for each scan for 1200 time points. The participants are at rest with eyes open with relaxed fixation on a projected bright cross-hair on a dark back-ground [72]. The images went through the standard minimal processing procedures explained in [74] such as spatial distortion removal [75], motion correction [76], bias field reduction [77], registration to the structural MNI template. This resulted in the resting-state functional time series with $91 \times 109 \times 91$, $2mm$ isotropic voxels at 1200 time points.

The scrubbing is further done to remove fMRI volumes with spatial artifacts in functional connectivity due to significant head motion [78,79]. The framewise displacement (FD) from the three translational displacements and three rotational displacements to measure the head movement from one volume to the next was calculated. The volumes with FD larger than $0.5mm$ and their neighbors were scrubbed [78,79]. About 12 subjects having excessive head movement were excluded resulting in 400 subjects (168 males and 232 females). Additional details on the dataset can be found here [79,80]. The subjects were in ages between 22 to 36 years (average age 29.24 ± 3.39 years). The Automated Anatomical Labeling (AAL) template was applied to parcellate the brain volume into 116 regions [81]. The fMRI across voxels within each brain parcellation is averaged, which produced 116 average fMRI time series with 1200 time points for each subject [44,79]. The Pearson correlation matrix $\rho = (\rho_{ij})$ was computed over the whole time points across 116 brain regions resulting in 400 correlation matrices of size 116×116 . Since the dataset contains $p = 116$ nodes, the total number of edges in the brain network is $q = p(p - 1)/2 = 6670$.

4.2 Hodge decomposition of the brain network

Following the proposed method, we decomposed individual brain networks using the Hodge decomposition. Figure 14 displays the average decomposition of 400 subjects. We first assessed if there are topological difference between females and males in the original connectivity (edge flow). Following the test procedure outlined in Section 2.5, the Wasserstein distances \mathcal{L}_∞^b on birth values for testing 0D topology difference and \mathcal{L}_∞^d on death values for testing 1D topology difference are separately used. The permutation test is conducted for the birth set (first term) and death set (second term) separately on 100000 permutations. We obtained the p-values 0.0177 and 0.0110 respectively showing there are topological difference present in the original connectivity matrices.

We further determined if we can detect topological differences in the decomposed components. There are several interesting properties of the Hodge decomposition that motivate us to study each component. The gradient component sum to zero along any cycles. The curl components are zero for edges that are not a 2-simplex boundary and the entries sum to zero around each node. The harmonic component sums to zero around each node, and it also sums to zero along each 2-simplex. The edge values on the gradient component show smaller values near the diagonal and higher edge values far away from the diagonal. The curl component shows higher values along the diagonal. We checked if the loop ratio (13) can be used to discriminate the brain networks between females and males. Figure 15 shows the distinct distribution differences in the loop ratio and corresponding empirical cumulative distributions. The plots suggests that there are more number of loops corresponding to the smaller edge weights in the males than in the females. We tested the topological equivalence

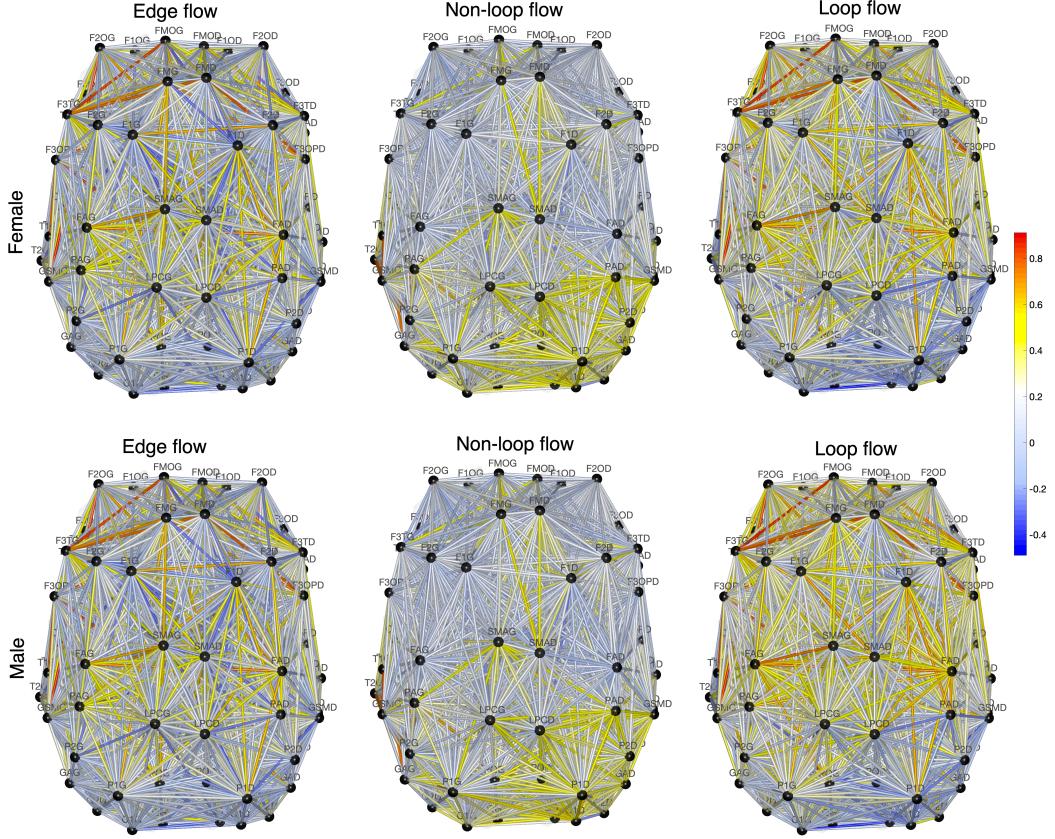


Fig. 14: Top: The average connectivity (edge flow), non-loop flow (middle) and the loop flow (right) of the female (top) and male networks (bottom).

of female brain networks and male brain networks using the Wasserstein distance (25). We first considered the network constructed from the gradient component and performed the birth death decomposition. We then carried out the permutation test with 100000 permutations ($p\text{-value} = 0.008$). Also we performed the permutation test on the curl component ($p\text{-value} = 0.0296$). Figure 16 shows the distribution of the test statistic with the observed statistic marked as dashed line. We conclude that the non-loop and loop components were able to detect the group difference.

We further tested if we can detect 0D and 1D topological signals separately in each component. Following the test procedure outlined in Section 2.5, the Wasserstein distances \mathcal{L}_∞^b on birth values for testing 0D topology difference and \mathcal{L}_∞^d on death values for testing 1D topology difference are separately used in each component. For non-loop component, the permutation test with 100000 permutations give the $p\text{-value}$ of 0.0088 and 0.0080 for the birth (0D topology) and and death (1D topology) values respectively. For loop component, the permutation test with 100000 permutations give the $p\text{-value}$ of 0.0019 and 0.0019 for the birth (0D topology) and and death (1D topology) values respectively. We conclude that each composition can be used for detecting 0D and 1D topological signals.

5 Conclusion

In this work, we explained how to extract the orthogonal components of the Hodge decomposition: gradient, curl and harmonic components and use them for topological inference in discriminating two groups of brain networks. We performed the Hodge decomposition

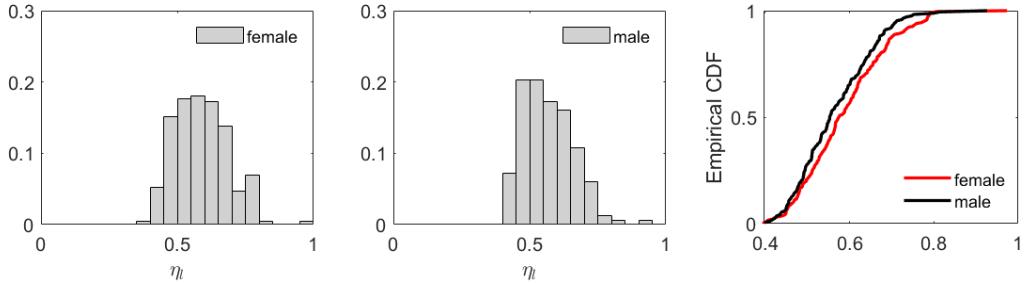


Fig. 15: The histogram showing the distribution of the loop ratio for female (left) and male (middle) subjects. The empirical cumulative distribution suggests there are more number of loops corresponding to the smaller edge weights in the males than in the females.

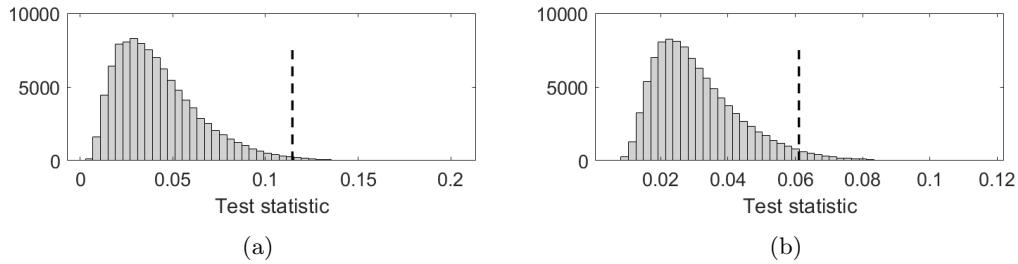


Fig. 16: The histogram of Wasserstein distance statistic for the gradient component with the observed statistics 0.1145 is marked as the dashed line. (b) The histogram of the Wasserstein distance statistic for the curl component with the observed statistics 0.061 is marked as the dashed line.

on correlation networks and obtained the gradient, curl and harmonic components of the decomposition. In the Hodge decomposition, the gradient component is computed from the potential function function defined on the nodes (1-simplices) while the curl component is obtained from the vector potential defined on the triangles (2-simplices). We then grouped the components as loop and non-loop flows. The Wasserstein distance based topological inference procedure was shown to provide the optimal matching and alignment at the edge level. Further, we used the ∞ -Wasserstein distance and incorporate topological features (birth and death values) of the loop flows and non-loop flows to discriminate networks. These studies indicated that the components contain topological features that can detect topological similarity and dissimilarity. The use of the Wasserstein distance based maximum gap statistic avoids the need for multiple comparisons and provide a single summary p-value as a measure of network difference.

The proposed method was applied to human brain networks obtained from fMRI dataset and showed that they can differentiate male and female brain networks. Although we have used Hodge decomposition on static functional brain networks in this study, the method can be easily extended to time varying graphs to analyze the temporal dynamics. It would be an interesting exercise to examine the diffusion behavior, synchronization and topological features on time dependent higher order networks. We leave these challenges as a future study.

Acknowledgements

We thank Kelin Xia of Nanyang Technological University and Sixtus Dakurah for discussion and support of the project. We also like to thank Shih-Gu Huang of National University

of Singapore for providing support for fMRI processing. This study is funded by NIH R01 EB022856, EB02875, NSF MDS-2010778.

References

1. P. J. Uhlhaas and W. Singer, “Neural synchrony in brain disorders: relevance for cognitive dysfunctions and pathophysiology,” *Neuron*, vol. 52, no. 1, pp. 155–168, 2006.
2. P. J. Uhlhaas, G. Pipa, B. Lima, L. Melloni, S. Neuenschwander, D. Nikolić, and W. Singer, “Neural synchrony in cortical networks: history, concept and current status,” *Frontiers in Integrative Neuroscience*, vol. 3, p. 17, 2009.
3. C. J. Stam and J. C. Reijneveld, “Graph theoretical analysis of complex networks in the brain,” *Nonlinear Biomedical Physics*, vol. 1, no. 1, pp. 1–19, 2007.
4. E. T. Bullmore and O. Sporns, “Complex brain networks: graph theoretical analysis of structural and functional systems,” *Nature Reviews Neuroscience*, vol. 10, no. 3, pp. 186–198, 2009.
5. E. T. Bullmore and D. S. Bassett, “Brain graphs: graphical models of the human brain connectome,” *Annual Review of Clinical Psychology*, vol. 7, pp. 113–140, 2011.
6. X. N. Zuo, R. Ehmke, M. Mennes, D. Imperati, F. X. Castellanos, O. Sporns, and M. P. Milham, “Network centrality in the human functional connectome,” *Cerebral Cortex*, vol. 22, no. 8, pp. 1862–1875, 2012.
7. M. E. Newman, “Finding community structure in networks using the eigenvectors of matrices,” *Physical Review E*, vol. 74, no. 3, p. 036104, 2006.
8. O. Sporns, C. J. Honey, and R. Kötter, “Identification and classification of hubs in brain networks,” *PloS One*, vol. 2, no. 10, p. e1049, 2007.
9. M. P. van den Heuvel and O. Sporns, “Network hubs in the human brain,” *Trends in Cognitive Sciences*, vol. 17, no. 12, pp. 683–696, 2013.
10. C. Giusti, R. Ghrist, and D. S. Bassett, “Two’s company, three (or more) is a simplex,” *Journal of Computational Neuroscience*, vol. 41, no. 1, pp. 1–14, 2016.
11. F. Battiston, G. Cencetti, I. Iacopini, V. Latora, M. Lucas, A. Patania, J.-G. Young, and G. Petri, “Networks beyond pairwise interactions: structure and dynamics,” *Physics Reports*, vol. 874, pp. 1–92, 2020.
12. P. Skardal and A. Arenas, “Higher order interactions in complex networks of phase oscillators promote abrupt synchronization switching,” *Communications Physics*, vol. 3, pp. 1–6, 2020.
13. D. V. Anand, S. Dakurah, B. Wang, and M. K. Chung, “Hodge-laplacian of brain networks and its application to modeling cycles,” *arXiv preprint arXiv:2110.14599*, 2021.
14. A. E. Sizemore, J. E. Phillips Cremins, R. Ghrist, and D. S. Bassett, “The importance of the whole: topological data analysis for the network neuroscientist,” *Network Neuroscience*, vol. 3, no. 3, pp. 656–673, 2019.
15. A. E. Sizemore, C. Giusti, A. Kahn, J. M. Vettel, R. F. Betzel, and D. S. Bassett, “Cliques and cavities in the human connectome,” *Journal of computational neuroscience*, vol. 44, no. 1, pp. 115–145, 2018.
16. H. Edelsbrunner and J. Harer, *Computational Topology: an introduction*. American Mathematical Society., 2010.
17. M. K. Chung, V. Villalta Gil, H. Lee, P. J. Rathouz, B. B. Lahey, and D. H. Zald, “Exact topological inference for paired brain networks via persistent homology,” in *International Conference on Information Processing in Medical Imaging*, pp. 299–310, Springer, 2017.
18. J. J. Torres and G. Bianconi, “Simplicial complexes: higher-order spectral dimension and dynamics,” *Journal of Physics: Complexity*, vol. 1, no. 1, p. 015002, 2020.
19. H. Edelsbrunner, J. Harer, et al., “Persistent homology-a survey,” *Contemporary Mathematics*, vol. 453, pp. 257–282, 2008.
20. G. Carlsson, “Topology and data,” *Bulletin of the American Mathematical Society*, vol. 46, no. 2, pp. 255–308, 2009.
21. M. K. Chung, H. Lee, A. DiChristofano, H. Ombao, and V. Solo, “Exact topological inference of the resting-state brain networks in twins,” *Network Neuroscience*, vol. 3, no. 3, pp. 674–694, 2019.
22. M. K. Chung, V. Vilalta Gil, P. J. Rathouz, B. B. Lahey, and D. H. Zald, “Mapping heritability of large-scale brain networks with a billion connections via persistent homology,” *arXiv preprint arXiv:1509.04771*, 2015.

23. M. K. Chung, H. Lee, V. Solo, R. J. Davidson, and S. D. Pollak, “Topological distances between brain networks,” in *International Workshop on Connectomics in Neuroimaging*, pp. 161–170, Springer, 2017.
24. Y. Wang, H. Ombao, and M. K. Chung, “Topological data analysis of single-trial electroencephalographic signals,” *The Annals of Applied Statistics*, vol. 12, no. 3, p. 1506, 2018.
25. R. Ghrist, “Barcodes: the persistent topology of data,” *Bulletin of the American Mathematical Society*, vol. 45, no. 1, pp. 61–75, 2008.
26. Z. Meng and K. Xia, “Persistent spectral-based machine learning (PerSpect ML) for protein-ligand binding affinity prediction,” *Science Advances*, vol. 7, no. 19, p. eabc5329, 2021.
27. X. Jiang, L.-H. Lim, Y. Yao, and Y. Ye, “Statistical ranking and combinatorial hodge theory,” *Mathematical Programming*, vol. 127, no. 1, pp. 203–244, 2011.
28. L. H. Lim, “Hodge Laplacians on graphs,” *SIAM Review*, vol. 62, no. 3, pp. 685–715, 2020.
29. M. T. Schaub, A. R. Benson, P. Horn, G. Lippner, and A. Jadbabaie, “Random walks on simplicial complexes and the normalized Hodge 1-Laplacian,” *SIAM Review*, vol. 62, no. 2, pp. 353–391, 2020.
30. C. Ziegler, P. S. Skardal, H. Dutta, and D. Taylor, “Balanced hodge laplacians optimize consensus dynamics over simplicial complexes,” *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 32, no. 2, p. 023128, 2022.
31. H. Lee, M. K. Chung, H. Kang, and D. S. Lee, “Hole detection in metabolic connectivity of Alzheimer’s disease using k - Laplacian,” in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 297–304, Springer, 2014.
32. S. Krishnagopal and G. Bianconi, “Spectral detection of simplicial communities via hodge laplacians,” *Physical Review E*, vol. 104, no. 6, p. 064303, 2021.
33. C. Bick, E. Gross, H. A. Harrington, and M. T. Schaub, “What are higher-order networks?,” *arXiv preprint arXiv:2104.11329*, 2021.
34. S. Barbarossa and S. Sardellitti, “Topological signal processing over simplicial complexes,” *IEEE Transactions on Signal Processing*, vol. 68, pp. 2992–3007, 2020.
35. T. Haruna and Y. Fujiki, “Hodge decomposition of information flow on small-world networks,” *Frontiers in neural circuits*, vol. 10, p. 77, 2016.
36. R. Zhao, M. Desbrun, G. W. Wei, and Y. Tong, “3d hodge decompositions of edge-and face-based vector fields,” *ACM Transactions on Graphics (TOG)*, vol. 38, no. 6, pp. 1–13, 2019.
37. H. Bhatia, G. Norgard, V. Pascucci, and P.-T. Bremer, “The helmholtz-hodge decomposition—a survey,” *IEEE Transactions on visualization and computer graphics*, vol. 19, no. 8, pp. 1386–1404, 2012.
38. T. McGraw, T. Kawai, I. Yassine, and L. Zhu, “Visualizing high-order symmetric tensor field structure with differential operators,” *Journal of Applied Mathematics*, vol. 2011, 2011.
39. M. K. Chung, J. L. Hanson, J. Ye, R. J. Davidson, and S. D. Pollak, “Persistent homology in sparse regression and its application to brain morphometry,” *IEEE Transactions on Medical Imaging*, vol. 34, no. 9, pp. 1928–1939, 2015.
40. H. Edelsbrunner, D. Letscher, and A. Zomorodian, “Topological persistence and simplification,” in *Proceedings 41st Annual Symposium on Foundations of Computer Science*, pp. 454–463, IEEE, 2000.
41. H. Lee, H. Kang, M. K. Chung, B. N. Kim, and D. S. Lee, “Persistent brain network homology from the perspective of dendrogram,” *IEEE Transactions on Medical Imaging*, vol. 31, no. 12, pp. 2267–2277, 2012.
42. H. Lee, M. K. Chung, H. Kang, B.-N. Kim, and D. S. Lee, “Computing the shape of brain networks using graph filtration and Gromov-Hausdorff metric,” in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 302–309, Springer, 2011.
43. M. K. Chung, S. G. Huang, A. Gritsenko, L. Shen, and H. Lee, “Statistical inference on the number of cycles in brain networks,” in *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, pp. 113–116, IEEE, 2019.
44. T. Songdechakraiut and M. K. Chung, “Topological learning for brain networks,” *arXiv preprint arXiv:2012.00675*, 2020.
45. S. Das, D. V. Anand, and M. K. Chung, “Topological data analysis of human brain networks through order statistics,” *arXiv preprint arXiv:2204.02527*, 2022.
46. Y. Mileyko, S. Mukherjee, and J. Harer, “Probability measures on the space of persistence diagrams,” *Inverse Problems*, vol. 27, no. 12, p. 124007, 2011.
47. R. Kwitt, S. Huber, M. Niethammer, W. Lin, and U. Bauer, “Statistical topological data analysis-a kernel perspective,” *Advances in neural information processing systems*, vol. 28, 2015.

48. J. J. Berwald, J. M. Gottlieb, and E. Munch, “Computing Wasserstein distance for persistence diagrams on a quantum computer,” *arXiv preprint arXiv:1809.06433*, 2018.
49. Z. Su, W. Zeng, Y. Wang, Z.-L. Lu, and X. Gu, “Shape classification using wasserstein distance for brain morphometry analysis,” in *International Conference on Information Processing in Medical Imaging*, pp. 411–423, Springer, 2015.
50. W. Zhang, J. Shi, J. Yu, L. Zhan, P. Thompson, and Y. Wang, “Enhancing diffusion MRI measures by integrating grey and white matter morphometry with hyperbolic wasserstein distance,” in *2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017)*, pp. 520–524, IEEE, 2017.
51. J. Shi, W. Zhang, and Y. Wang, “Shape analysis with hyperbolic wasserstein distance,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5051–5061, 2016.
52. J. Shi and Y. Wang, “Hyperbolic wasserstein distance for shape indexing,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, pp. 1362–1376, 2019.
53. S. Gerber, M. Niethammer, E. Ebrahim, J. Piven, S. Dager, M. Styner, S. Aylward, and A. Enquobahrie, “Optimal transport features for morphometric population analysis,” *arXiv preprint arXiv:2208.05891*, 2022.
54. J. Yan, C. Deng, L. Luo, X. Wang, X. Yao, L. Shen, and H. Huang, “Identifying imaging markers for predicting cognitive assessments using wasserstein distances based matrix regression,” *Frontiers in Neuroscience*, vol. 13, p. 668, 2019.
55. J. R. Munkres, “Algorithms for the assignment and transportation problems,” *Journal of the Society for Industrial and Applied Mathematics*, vol. 5, no. 1, pp. 32–38, 1957.
56. Z. Su, Y. Wang, R. Shi, W. Zeng, J. Sun, F. Luo, and X. Gu, “Optimal mass transport for shape matching and comparison,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 11, pp. 2246–2259, 2015.
57. Z. Su, W. Zeng, Y. Wang, Z. L. Lu, and X. Gu, “Shape classification using Wasserstein distance for brain morphometry analysis,” in *International Conference on Information Processing in Medical Imaging*, pp. 411–423, Springer, 2015.
58. J. M. Steele, *The Cauchy-Schwarz master class: an introduction to the art of mathematical inequalities*. Cambridge University Press, 2004.
59. S. Das, H. Ombao, and M. K. Chung, “Topological data analysis for functional brain networks,” *arXiv preprint arXiv:2210.09092*, 2022.
60. J. Chen, R. Zhao, Y. Tong, and G. W. Wei, “Evolutionary de Rham-Hodge method,” *arXiv preprint arXiv:1912.12388*, 2019.
61. F. Fouss, K. Francoisse, L. Yen, A. Pirotte, and M. Saerens, “An experimental investigation of kernels on graphs for collaborative recommendation and semisupervised classification,” *Neural networks*, vol. 31, pp. 53–72, 2012.
62. B. C. Hall *et al.*, *Lie groups, Lie algebras, and representations: an elementary introduction*, vol. 10. Springer, 2003.
63. R. Tarjan, “Depth-first search and linear graph algorithms,” *SIAM Journal on Computing*, vol. 1, no. 2, pp. 146–160, 1972.
64. A. Muhammad and M. Egerstedt, “Control using higher order Laplacians in network topologies,” in *Proceedings of 17th International Symposium on Mathematical Theory of Networks and Systems*, pp. 1024–1038, Citeseer, 2006.
65. C. Chen and D. Freedman, “Measuring and computing natural generators for homology groups,” *Computational Geometry*, vol. 43, no. 2, pp. 169–181, 2010.
66. B. Bollobás and B. Béla, *Random graphs*. No. 73, Cambridge university press, 2001.
67. A. Frieze and M. Karoński, *Introduction to random graphs*. Cambridge University Press, 2016.
68. S. L. Simpson, S. Hayasaka, and P. J. Laurienti, “Exponential random graph modeling for complex brain networks,” *PloS one*, vol. 6, no. 5, p. e20039, 2011.
69. A. Fujita, M. C. Vidal, and D. Y. Takahashi, “A statistical method to distinguish functional brain networks,” *Frontiers in Neuroscience*, vol. 11, p. 66, 2017.
70. H. Evans and N. Hastings, “Peacock. statistical distributions,” 2000.
71. M. K. Chung, H. Lee, A. Gritsenko, A. DiChristofano, D. Pluta, H. Ombao, and V. Solo, “Topological brain network distances,” *arXiv preprint arXiv:1809.03878*, 2018.
72. D. C. Van Essen, K. Ugurbil, E. Auerbach, D. Barch, T. E. Behrens, R. Bucholz, A. Chang, L. Chen, M. Corbetta, S. W. Curtiss, *et al.*, “The Human Connectome Project: a data acquisition perspective,” *Neuroimage*, vol. 62, no. 4, pp. 2222–2231, 2012.

73. D. C. Van Essen, S. M. Smith, D. M. Barch, T. E. Behrens, E. Yacoub, K. Ugurbil, W.-M. H. Consortium, *et al.*, “The WU-Minn human connectome project: an overview,” *Neuroimage*, vol. 80, pp. 62–79, 2013.
74. M. F. Glasser, S. N. Sotiropoulos, J. A. Wilson, T. S. Coalson, B. Fischl, J. L. Andersson, J. Xu, S. Jbabdi, M. Webster, J. R. Polimeni, *et al.*, “The minimal preprocessing pipelines for the Human Connectome Project,” *Neuroimage*, vol. 80, pp. 105–124, 2013.
75. J. L. Andersson, S. Skare, and J. Ashburner, “How to correct susceptibility distortions in spin-echo echo-planar images: application to diffusion tensor imaging,” *Neuroimage*, vol. 20, no. 2, pp. 870–888, 2003.
76. M. Jenkinson and S. Smith, “A global optimisation method for robust affine registration of brain images,” *Medical Image Analysis*, vol. 5, no. 2, pp. 143–156, 2001.
77. M. F. Glasser and D. C. Van Essen, “Mapping human cortical areas in vivo based on myelin content as revealed by T1-and T2-weighted MRI,” *Journal of Neuroscience*, vol. 31, no. 32, pp. 11597–11616, 2011.
78. J. D. Power, K. A. Barnes, A. Z. Snyder, B. L. Schlaggar, and S. E. Petersen, “Spurious but systematic correlations in functional connectivity MRI networks arise from subject motion,” *Neuroimage*, vol. 59, no. 3, pp. 2142–2154, 2012.
79. S.-G. Huang, S. B. Samdin, C.-M. Ting, H. Ombao, and M. K. Chung, “Statistical model for dynamically-changing correlation matrices with application to brain connectivity,” *Journal of neuroscience methods*, vol. 331, p. 108480, 2020.
80. T. Songdechakraiwut, L. Shen, and M. Chung, “Topological learning and its application to multimodal brain network integration,” *Medical Image Computing and Computer Assisted Intervention (MICCAI)*, vol. 12902, pp. 166–176, 2021.
81. N. Tzourio Mazoyer, B. Landeau, D. Papathanassiou, F. Crivello, O. Etard, N. Delcroix, B. Mazoyer, and M. Joliot, “Automated anatomical labeling of activations in SPM using a macroscopic anatomical parcellation of the MNI MRI single-subject brain,” *Neuroimage*, vol. 15, no. 1, pp. 273–289, 2002.