AFRL-RI-RS-TR-2020-092

TOPOLOGICAL EXPLOITATION

MAY 2020

FINAL TECHNICAL REPORT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

STINFO COPY

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE

■ **AIR FORCE MATERIEL COMMAND** ■ **UNITED STATES AIR FORCE** ■ **ROME, NY 13441**

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report was cleared for public release by the 88th ABW, Wright-Patterson AFB Public Affairs Office and is available to the general public, including foreign nationals. Copies may be obtained from the Defense Technical Information Center (DTIC) (http://www.dtic.mil).

AFRL-RI-RS-TR-2020-092 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE CHIEF ENGINEER:

/ S /	**/ S /**
JOHN J. KELLY	JAMES S. PERRETTA
Work Unit Manager	Deputy Chief, Information
	Exploitation & Operations Division
	Information Directorate

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

REPORT DOCUMENTATION PAGE	*Form Approved* **OMB No. 0704-0188**

1. REPORT DATE *(DD-MM-YYYY)* MAY 2020	**2. REPORT TYPE** FINAL TECHNICAL REPORT	**3. DATES COVERED** *(From - To)* OCT 2015 – OCT 2019

4. TITLE AND SUBTITLE Topological Exploitation	**5a. CONTRACT NUMBER** IN-HOUSE
	5b. GRANT NUMBER N/A
	5c. PROGRAM ELEMENT NUMBER 62788F
6. AUTHOR(S) John Kelly	**5d. PROJECT NUMBER** RIGI
	5e. TASK NUMBER HT
	5f. WORK UNIT NUMBER OP

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Air Force Research Laboratory/Information Directorate Rome Research Site/RIGC 525 Brooks Road Rome NY 13441-4505	**8. PERFORMING ORGANIZATION REPORT NUMBER** N/A
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Air Force Research Laboratory/Information Directorate Rome Research Site/RIGC 525 Brooks Road Rome NY 13441-4505	**10. SPONSOR/MONITOR'S ACRONYM(S)** AFRL/RI
	11. SPONSORING/MONITORING AGENCY REPORT NUMBER AFRL-RI-RS-TR-2020-092

12. DISTRIBUTION AVAILABILITY STATEMENT
Approved for Public Release; Distribution Unlimited. PA# 88ABW-2020-0332
Date Cleared: 31 Jan 2020

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This report summarizes the results of the TOPEX project. The Topological Exploitation project (TOPEX) initially aimed at understanding and studying recent methods in topological data analysis (TDA). The research contained is intended to provide a basic under-standing of TDA and the type of signal information that TDA tools provide, with the aim of providing methods to facilitate unsupervised detection of structure within data. Additionally, additional tools are investigated to perform structure detection in signal processing and data analysis (e.g. covariance distance measures and singular value analysis). Lastly, some high-level applications are explored. For example, the utility of TDA for opportunistic data mining is briefly analyzed, along with some of the general issues relating to data mining.

15. SUBJECT TERMS
Aggregation/clustering; cellular sheaf/sheaves; sensor networks; sheaf theory; SIGNAL targeting; Tactical SIGINT; signal analysis

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** **JOHN J. KELLY**
a. REPORT U	**b. ABSTRACT** U	**c. THIS PAGE** U	UU	85	**19b. TELEPHONE NUMBER (*Include area code)*** **NA**

TABLE OF CONTENTS

List of Figures ii

1.0 Summary 1

2.0 Introduction 2

3.0 Methods 4

4.0 Results 26

5.0 Conclusion 47

6.0 References 48

APPENDIX A: DERIVATION OF EXPECTED FROBENIUS NORM METRIC 51

APPENDIX B: DISTRIBUTIONS OF OFF-DIAGONAL CORRELATION MATRIX . . 61

APPENDIX C: TOPOLOGICAL BACKGROUND 62

APPENDIX D: SHEEHY REPORT 66

List of Symbols 79

List of Abbreviations 80

List of Figures

1 Matrix decomposition approach used to detect structure in data. 7
2 High-Level Data Flow Diagram for Clique Topology 10
3 Detailed Data Flow Diagram for Clique Topology 10
4 Failure of Single Parameter Gram Matrix Measures 22
5 Initial Prototype Example-Single Run 26
6 Data Flow for Initial Prototype Example 27
7 Betti Curve Areas for 8 Correlated Subspaces 27
8 Betti Curve Areas for 4 Correlated Subspaces 28
9 Control Persistance Diagram 29
10 Persistence Diagrams of Thresholded EDMs 30
11 Persistence Diagrams of Handwritten Digits 31
12 Masks Developed for Images 32
13 Bottleneck Distance 33
14 Embedding of 1-Sphere 33
15 Embedding of 2-Sphere 34
16 Klein Bottle Embeddings 34
17 MDS of Persistence Diagrams (p_1,0) 35
18 MDS of Persistence Diagrams (p_1,3) 36
19 MDS of Persistence Diagrams (p_2,6) 36
20 MDS of Persistence Diagrams (p_3,2) 37
21 Informativeness with Bartlett's Method 39
22 Informativeness with Bures Method 40
23 Scree Plots for Various Levels of Structure 41
24 Random Hadamard Multiplication Structure Test Results 42
25 Off-Diagonal Element vs. Random Hadamard Structure Test 43
26 Shannon Entropy Structure Test Results 44
27 Frobenius Norm Structure Test Results 45
28 Kolmogorov-Smirnov Test Results 45
29 Record Length Impact on Frobenius Norm Covariance Distance 46
30 Basic Steps to Compute Homology 63
31 Covariance Matrix Cliques, Loops, and Orderings 65

1 Summary

This report summarizes the results of the TOPEX project. The Topological Exploitation project (TOPEX) initially aimed at understanding and studying recent methods in topolog-ical data analysis (TDA). The research contained is intended to provide a basic under-standing of TDA and the type of signal information that TDA tools provide, with the aim of providing methods to facilitate unsupervised detection of structure within data. Addi-tionally, additional tools are investigated to perform structure detection in signal process-ing and data analysis (e.g. covariance distance measures and singular value analysis). Lastly, some high-level applications are explored. For example, the utility of TDA for op-portunistic data mining is briefly analyzed, along with some of the general issues relating to data mining.

2 Introduction

2.1 Finding Structure within Data Sets

The task of data analysis is becoming increasingly necessary and difficult with the well-documented influx of collected data that is so common today. While the field of data analysis is very large and ever-growing, there are a growing set of tools within the field of topological data analysis (TDA). TDA has recently garnered increasing attention in the literature. During the literature search for this project, one TDA publication was identified, which applied TDA to study brain activity in the field of neuroscience. This method utilized a novel matrix analysis tool, called *clique topology* which was used to study the geometric structure of covariance and correlation matrices. It is this method of clique topology that forms the basis for the TDA approach studied in the TOPEX project. Within the current literature, there are many different research efforts directed at TDA for structure detection. In general, the focal task of these tools is towards unsupervised structure detection and data mining.

Data mining tasks include outlier identification, clustering, and pattern finding, i. e. finding a simplified model that matches the data. Related areas include *data cleansing* [1] and *denoising* [2]. In the context of classification, e.g. structure detection can be performed on residuals after classification has taken place, i.e., analyzing what's left over, after known structures in the data have been accounted for. Exploratory data analysis with regard to online structure detection is related to change detection, change-point detection and stationarity. The task of structure detection is really one of distinguishing between *organized* structure and *random* structure. Recently, this concept has been formalized in the notions of *informativeness* and *non-informativeness* [2] which aim at quantifying the homogeneity of a set of measurements. In this effort, the aim has been to identify tools that can aid in automated screening of sensor network data for interesting structure.

With regards to data mining, *controlled experiments*, are specifically designed to test the veracity of a hypothesis, i.e. whether or not a stated expected result is achieved [3]. The aim of the test is usually directed at ascertaining the cause of of some observed stimuli through demonstrating or proving that there exists a dependent relationship between the controlled input parameters and the resulting observations or measurements. This requires careful collection and handling of the data acquisition process, as well as specific manipulation of the circumstances under which the hypothesis is tested [3], [4].

In contrast to controlled experiments, *natural experiments* make use of *data of opportunity*, i.e. data that happens to present itself through some convenient means, and allow one to test claims that may be possible otherwise [3], [4]. While natural experiments result in less control over the experiment (due e.g., to ethical, logistical, legal, etc. issues), they can provide information not otherwise obtainable. However, the justification between cause and effect required for elevating a factor from dependency to cause is generally more involved in such natural experiments, as opposed to controlled experiments [3], [4]. The tools developed in this effort are mainly aimed at benefiting situations where analysts have less control over the experiment. In such circumstances, tools from unsupervised data mining, pattern recognition and exploratory data analysis can be applied.

2.2 Organization of This Report

This report is organized as follows. First, Section 3 provides an overview of the meth-ods that were studied to perform data analysis. While these methods include TDA, other methods were studied as well. The theory and some references to the literature are pro-vided in this section. Next, Section 4 provides some results from prototyping the methods described in Section 3. Finally, Section 5 lists some general potential commercial appli-cations, along with a summary and some areas for possible future research.

3 Methods

3.1 Overview

3.1.1 General Approach: Matrix Decompositions

Large data sets are unwieldy to work with, so it of computational benefit to summarize the data set by reducing the dimensionality of the data set, either by decomposing the data set into subsets of simpler constituent components or transforming the data to re-express it such that features of interest become more apparent. Then the subset of data that are deemed more significant or more relevant can be further analyzed, instead of analyzing the entire data set. Within this effort, the focus was on decomposing the data (or a func-tion of the data) into intrinsically simpler pieces or components. Matrix decompositions can be used for a wide range of purposes, many of which result in a simplification of the data set. However, two very popular uses are to perform either denoising or clustering of the data.

Inherent within the undertaking of dimensionality reduction (DR) is the hope that although the measurement vectors (where a single vector measurement record captures multiple dimensions) are taken over many times or many locations, the amount of information that is really important resides within a space of much lower dimensionality than the ambient dimension. I.e., the information lives within a small number of dimensions; small relative to the total number of dimensions over which the data is collected. Yet another way of expressing this is through the notion that the relevant information is *sparse* with regards to the measured data. Note that it may be that the data matrix itself is not natively sparse, but may be transformed such that the matrix is sparse in some other domain. Whether the data matrix is sparse or not, the important information contained within the data is typically sparse.

While there are many different DR tools, there are two basic types: linear and nonlinear. Well-known linear DR tools such as SVD, EVD, and PCA rely on a linear matrix decompositions to isolate significant or interesting linear subspaces. Linear matrix decompositions abound and include the SVD, QR, SDD, EVD, ICA, and NNMF [3].

3.1.2 Graph Interpretation of Matrix Decompositions

There are multiple interpretations of matrix decompositions [3]. One interpretation is that of a graph-theoretic or network perspective. Network science is a fairly recent approach to multi-dimensional data analysis and to signal processing, and represents data using nodes and vertices on a graph. An entire topological framework for performing signal processing over graphs continues to be developed [5], [6], [7]. While graph-theoretic ap-proaches have proven very useful, one drawback is that graphs encode edge relationships between **pairs** of nodes or dyads [8]. Higher-order relationships, e.g. between three or more nodes are generally not utilized.

3.1.3 Signal Model: Symmetric, Positive Semi-Definite Matrices

The analysis of sensor network data can take many forms. One common analytic object for sensor network data is that of an *affinity matrix*. An affinity matrix can be one of many specific types, e.g., Gram, covariance, correlation, adjacency, Laplacian, etc. depending on the application, and the viewpoint taken in the data analysis.

The objects under study here are mainly correlation, covariance, and Gram matrices, all of which have a symmetric positive semi-definite matrix (PSD) structure. More generally, these types of matrices have also been referred to as *affinity matrices* [3]. The data analyzed by the tools in this effort is assumed to be lumped into one single data collection, as opposed to having to re-analyze data after a series of sequential updates. For the latter, sub-space tracking methods may be used.

The NxN empirical covariance matrix C of an MxN data set X is computed as [9], [2]:

$$C = \frac{1}{N} X^T J_N X, \tag{1}$$

where the NxN matrix J_N is referred to as the *column centering matrix* and defined as

$$J_N = I_N - \frac{1}{N} 1_N 1_N^T, \tag{2}$$

where $1_N = [1\ 1, \ldots 1]^T \in \mathbb{R}^{Nx1}$. To convert the covariance (or any symmetric matrix) $\tilde{C}$ to a correlation matrix C, *symmetric normalization* can be performed as $C = D\tilde{C}D$, where $D_{i,i} = 1/\sqrt{\tilde{C}_{i,i}}$, ensuring that $C_{i,i} = 1$ [2].

3.1.4 Organization of This Chapter

Sub-section 3.1 provides a brief overview of TDA and an area known as *clique topol-ogy* is reviewed that studies the nearness of data points by quantifying the clustering or nearness of the points in a *clique*, or an all-to-all connected graph. Next, 3.1 persistence diagrams are analyzed under the condition that the relevant pairwise measurements are missing or known to be faulty and thus discarded. Next, the possibility of applying the output of persistent homology (a persistence diagram) as an input to a neural network for classifying the diagrams, thereby performing classification of the underlying data sets. Then, homological methods are studied and extended as applied to the problem of sen-sor network measurement data falsification. Then, sub-section 3.2 discusses some more detailed theory and problems related to TDA. Sub-section 3.3 describes the theory and rationale behind the topological prototyping experiments that were performed. In sub-section 3.4.7 a departure from the more topological approaches are explored. There, methods to measure distances between correlation, Gram, and covariance matrices are described. Lastly, sub-section 3.4 provides a collection of additional methods in the liter-ature that have been useful in structure detection.

3.2 TDA Literature Survey

3.2.1 Computing Topological Features of Graphs

This section presents some background on topological data analysis. Please see the ap-pendix *Topological Background* for additional foundational topics such as topology and homology. As a way to capture higher-order relationships in graphs, computationally effi-cient tools have recently been developed based on the field of *algebraic topology*. Such computational topology tools have been applied to data analysis and provide more de-tailed information about the non-linear characteristics of data, giving rise to the field of *topological data analysis* (TDA). As an example, one computational tool developed is a generalization of a graph representation, known as a *simplicial complexes*, which encode higher order relationships among nodes of a graph [8]. Historically signal processing and data analysis tools make extensive use of tools from computational geometry, measuring distance and lengths of signals. As sensors have become more prevalent and databases have grown, the need for more advanced analysis and multi-dimensional and heteroge-neous signal processing tools has grown as well. The data has an inherent geometric shape, and as the number of dimensions grows, there arises a need to study the large scale features of the data's shape such as the holes or voids in the data. Computational topology makes use of practical algorithms for computing topological measures such as connectedness, to analyze the shape of a data set. One way to measure connectedness is to count the number of connected components, holes or voids in the shape of a data set.

Alternatively stated, classical geometry-based signal processing tools focus on *geometric spaces*, and topological tools focus on analyzing more general topological features within *topological spaces*. A topological *invariant* or topological *property* of a surface is one that depends only on the topology of the surface; the property will not change with bending, twisting, or stretching [10]. The premise behind many existing signal processing tools is that data has geometric properties. Many geometric properties have been exploited in DSP research; indeed he entire notion using Euclidean vector spaces to represent and analyze signals implies the geometric concepts of distances and angles among signals.

Within the field of classic network science (or network theory), graphs encode dyadic relationships (i.e. edge relationships between no more than two nodes). Effective topological data analysis generally requires the data to be represented in such a way so as to capture more than just dyadic edge relationships between nodes. One way to represent higher-order notions such as surfaces (as opposed to just edges and nodes) is to use a simplicial complex, broadening the dyadic concepts from graph theory [8]. A simplicial complex is a generalization of the notion of a graph which encodes higher order relationships beyond that of graphs. The simplicial complex is a complex or combinatorial collection of simple components or simplices.

A filtration is a way to study simplicial complexes by decomposing them into simpler pieces, and analyzing each component separately, where each component resides at a different scale. Each component is a nested superset of the previous components, in the sense that they contain the previous components. Filtrations can also be used to threshold any weighted graph, even if only geometrical properties are analyzed. The mainstream approach to computational topology is to incrementally ’grow’ a data set by analyzing topological features across various scales; i.e. to perform a *filtration*. Filtrations robustly track changes in measures such as connectedness as the shape is stretched or twisted [11]. The longer a feature persists over the scale, the more topologically significant the feature is.

3.2.2 Persistent Homology: Tracking Topological Features Across Scales

Persistent homology can be defined as an approach to computing topological features (specifically homology) as they are born, persist and die across various scales [12]. The purpose of this effort was to explore and experiment with approaches from topological data analysis (TDA, specifically persistent homology) and compare and contrast it with some more traditional non-TDA approaches, e.g. matrix decompositions [3], (p. 27).

In [3], four interpretations of matrix decompositions are described including factor, geometric, component, and graph interpretations. The clique topology approach studied in this effort is a generalization of the graphical interpretation of the data. The clique topology approach decomposes can be viewed as a type of matrix decomposition. A correlation matrix decomposed into a series of nested binary matrices, by iteratively thresholding the correlation matrix, lowering the threshold and setting only those correlation elements above the current threshold to one and the remaining elements to zero. Then, each thresholded binarized matrix is used to compute homology. The homology is computed at each threshold, i.e. at each filtration level, and tracked as the threshold applied to the correlation matrix is decreased. In this way, the relationship between topology and geometry is explored, by iteratively growing the space while topological changes are computed [11].

The geometry of the space, i.e. the angles between each of the measurement vectors, forms the initial structure of the space. Then, spheres are inflated around each point, and connections are made between nodes on the basis of whether or not spheres intersect. The geometry is explored by encoding the geometry into the topological history, found by computing the filtration. Regarding using the correlation matrix as opposed to the operating on the actual data directly, one benefit is that depending on the dimensions of the data set analyzed, the dimensionality might be much smaller. A downside to this is that you either get information about the records (rows), or about the features (columns), but not both [3].

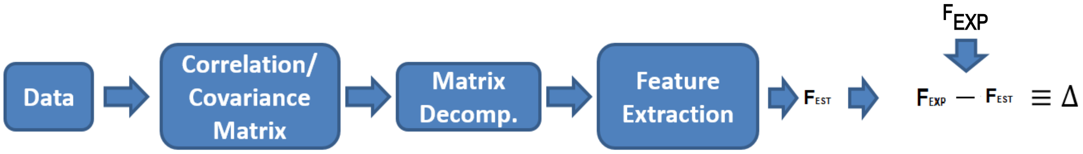

Data
Correlation/ Covariance Matrix
Matrix Decomp.
Feature Extraction
F_{EST}
F_{EXP}
$F_{EXP} - F_{EST} \equiv \Delta$

Figure 1: Matrix decomposition approach used to detect structure in data.

The focus of this work is to decompose the correlation or covariance matrix into one or more sub-matrices, then extract a feature (or set of features) as shown in Fig. 1. For example, the distance between two covariances could be used as the feature extracted from a data set (or collection of data sets). Then the expected feature (covariance distance in this example) F_{EXP} could be compared with the estimated covariance distance F_{EST}, to obtain a difference between the two.

3.2.3 Focus: Topological Detection of Geometric Structure

Within the literature, topological approaches to data analysis have become increasing popular in recent years [11]. Specifically, what is meant by a topological approach to signal processing (and more generally to data analysis) is to apply the knowledge that measured or collected data inherently has shape. To either store or process efficiently, it is often useful to first compress the data, e.g. in [13], then process the data to extract information.

Adapting from [11], TDA is interested in capturing and understanding the shapes of spaces. This understanding is really in the form of classifications: We would like to know how spaces agree and differ in shape in order to categorize them. There are two very general classes of properties of spaces: *extrinsic* and *intrinsic*. To do effectively categorize spaces, intrinsic properties of spaces need to be identified. "'We can try transforming a space in some fixed way and observe the properties that do not change. We call these properties the *invariants* of the space. Felix Klein gave this famous definition for geometry in his Erlanger Programm address in 1872"' [10].

For example, Euclidean geometry refers to the study of invariants under rigid motion, e.g., moving a cube in space does not change its geometry. Topology, on the other hand, studies invariants under continuous, and continuously invertible, transformations. For example, we can mold and stretch a play-doh ball into a filled cube by such transformations, but a ball cannot be stretched into a donut shape without cutting a hole in it [10].

3.2.4 Topology and Topological Data Analysis

The text [11] is a reference that provides a solid introduction to computational topology concepts, specifically, how topological concepts are defined, and their relationship to tasks in data analysis. The book [10], while somewhat mathematical in its approach, places a decent amount of emphasis on the history of geometry and the events that lead to the development of topology as we know it today. It provides a very helpful semi-technical backdrop to understand topology and homology in particular.

3.2.5 Clique Topology

In this research, the main topological method applied is based on that of [12] for the de-tection of geometric structure. In [12], the application was aimed at detecting structure in the neuronal activity of hippocampi within rat brains. The neuronal activity was recorded and correlation matrices among each pair of neurons was computed. Each entry of the correlation matrix was quantized to a single bit (zero or one), with respect to a series of thresholds. The result is what is known as a *filtration* in the literature on topological data analysis.

As pointed out in [12], an early applications of algebraic topology to neural data was (Singh et al, 2008). There, distributions of algebraic-topological features were used to discern if a macaque monkey was engaged in spontaneous activity, or was being exposed to natural images. The topological features used were *homological cycles*. Cycles give insight into global and hierarchical structure of the data, and therefore the global structure of the system from which the data was derived. These topological tools are also highly related to more general problems in *sensor networks*. Additionally, graph cliques are also studied in [14].

A high-level, qualitative algorithm for clique topology that is based on [12] is provided below:

1. Calculate the covariance matrix, which provides (minimally) a measure of the linear dependence among a set of vectors (columns of a matrix, atoms in a dictionary).

2. Induce an order complex (poset topology) onto the upper-half of the covariance matrix. This orders the elements in the covariance matrix by significance.

3. Represent the covariance matrix as a simple (undirected) graph. Then, after the application of the order complex, the order complex of the simple graph is represented as a sequence of nested graphs, as a function of increasing edge density.

4. For each graph in the sequence of nested graphs, compute the clique complex, the set of all cliques that is closed under the operation of taking subsets (under inclusion).

5. Sort these cliques by dimension (i.e. number of elements in each clique).

6. For each set of cliques, determine the boundaries of each clique in the set. Those cliques which have boundary of zero (i.e. the kernel of the boundary) are cycles.

7. *Quotient out* those cycles that are boundaries of higher-dimensional cycles. The intent is to avoid counting cycles twice, i.e. don't to distinguish between two cycles if one cycle can be deformed into the other cycle without leaving the clique complex.

8. At a given edge density, the homology of the clique complex provides a relationship among the cliques in the graph.

A clique is a subset of vertices of an undirected graph such that its induced subgraph is complete [12]. The *clique complex* X(G) of an undirected graph G is an abstract simplicial complex (that is, a family of finite sets closed under the operation of taking subsets), formed by the sets of vertices in the cliques of G. Any subset of a clique is itself a clique, so this family of sets meets the requirement of an abstract simplicial complex that every subset of a set in the family should also be in the family [12]. An *m-clique* consisting of *m* vertices, and a chain is a linear combination of cliques. Whereas Fig. 2 provides a high-level qualitative description of the general clique topology pipeline, Fig. 3 provides a more detailed view of the clique topology data flow.

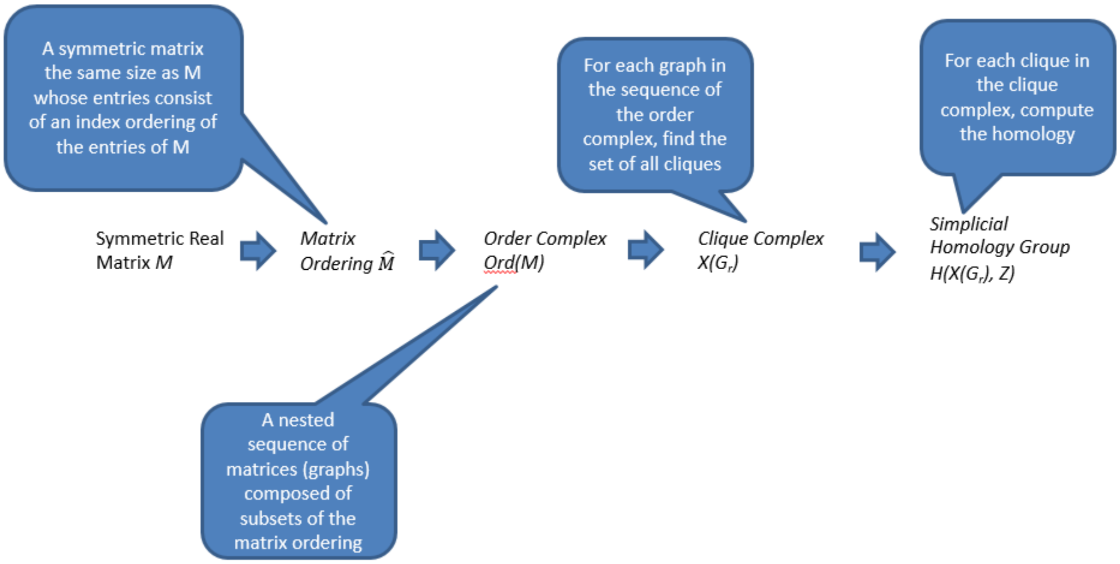

A symmetric matrix the same size as M whose entries consist of an index ordering of the entries of M
For each graph in the sequence of the order complex, find the set of all cliques
For each clique in the clique complex, compute the homology
Symmetric Real Matrix M
Matrix Ordering $\widehat{M}$
Order Complex $Ord(M)$
Clique Complex $X(G_r)$
Simplicial Homology Group $H(X(G_r), Z)$
A nested sequence of matrices (graphs) composed of subsets of the matrix ordering

Figure 2: High-Level Data Flow Diagram for Clique Topology

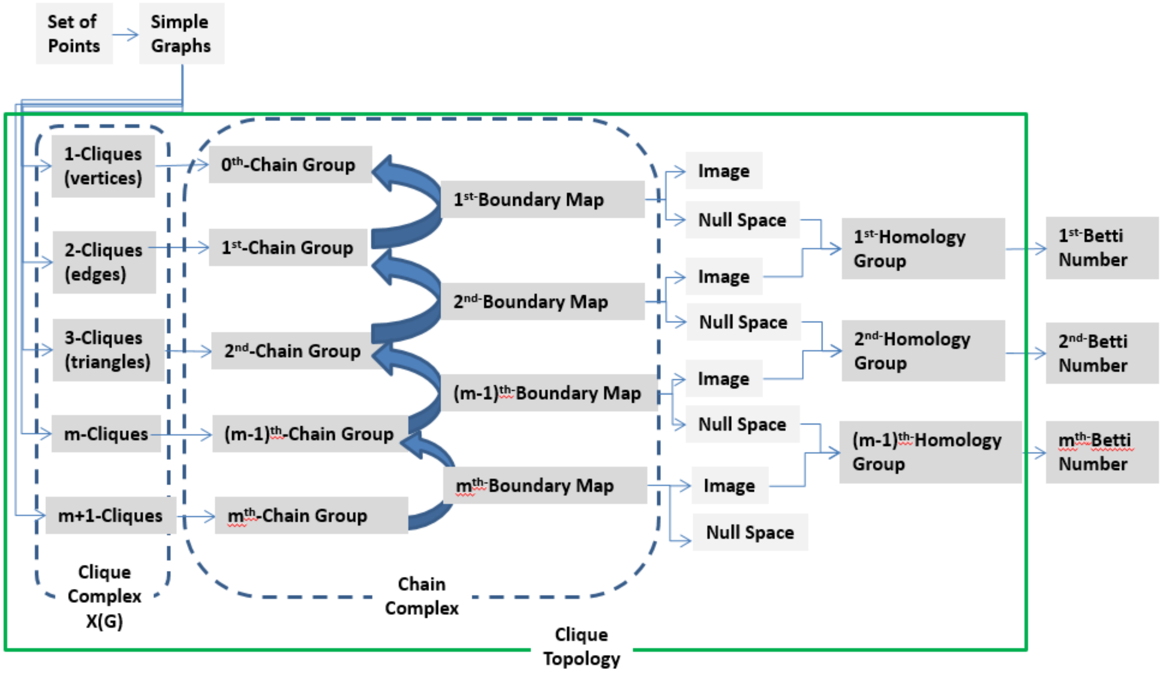


Figure 3: Detailed Data Flow Diagram for Clique Topology

The common practice of using standard tools such as eigenvalue decomposition is not appropriate for neural applications. Such matrix analysis relies on quantities which are invariant under linear change of basis, e.g. eigenvalue decomposition of a covariance matrix as in PCA are commonly employed [12]. This strategy is natural in the physical sciences, where meaningful quantities are frequently assumed to be preserved by linear coordinate transformations [12]. However, in contrast to common approaches, researchers believe that structure in neural data should be invariant under matrix transformations of the form $C = f(A)$ [12]. The eigen-structure of symmetric matrices such as covariances are distorted when applied to non-linear monotonically increasing transformations f(x) [12], instead "'The arrangement of cliques (all-to-all connected subgraphs) in the order complex of a matrix can be used in lieu of eigenvalues to detect random or geometric structure"' [12].

3.2.6 Structure and Change Detection in Correlation Matrices

Reference [15] proposed a method to detect change points in correlation networks for time series data that is not based on a specific type of distribution. The work in [15] merely focused on taking the Frobenius norm of the difference between two correlation matrices. However, there are many more norms that can be applied to detect structure change

3.3 Topological Approaches

3.3.1 Structure Detection Using Betti Curve Area

The first experiment with clique topology was performed by downloading the freely and publicly available "'CliqTop"' MATLAB code available from Dr. Giusti's website at

$$http://www.chadgiusti.com/tda-software.html.$$

The structured input data that was used for these experiments was randomly generated by randomly selecting a entries with single-mode, univariate, Gaussian distributions, and then adding additive white Gaussian noise (AWGN) to bring the signal to noise ratio (SNR) down to 15 dB.

The data matrix D was either a structured (the Gaussian distributed data with AWGN) or an unstructured (consisting of a shuffled version of the noiseless structured case) data set. These two types of data sets were analyzed by computing their respective covariance matrices, and then this covariance matrix C was fed into the clique topology algorithm. The output of the clique topology algorithm are the first β_1, second β_2, and third β_3 Betti numbers.

3.3.2 Persistence with Incomplete Pairwise Measurements

3.3.2.1 Introduction

One of the original motivations for the use of persistent homology in the study of data is the *stability theorem*. Roughly, the stability theorem says that the persistent homology of the filtered Vietoris-Rips complex built from a point cloud is robust to slight perturbation of the points.

In contrast, it is straightforward to construct examples for which classical (i.e., non-parameterized) homology changes drastically under small perturbation. These results have since been extended to include sub-sampling the points and noise in complexes built from non-metric data, like correlations between signals. However, none of this work considers the effect on persistence of gaps in the pairwise similiarity matrix used to construct the complex.

Such gaps appear commonly in real data, either as a result of a missing, corrupted, or untrustworthy measurement, or as a result of intentional sparsification or subsampling of the similarity matrix. Unfortunately, in this setting, it is possible for even small amounts of missing data to create dramatic changes in the resulting persistence diagram. Given the correspondence between pairwise measurements and filtration levels in which an edge appears in the standard filtered complex constructions, removing a single edge from the filtration can cause a cycle that has a finite death time to persist until the end of the filtration. In this case, the distance between the two persistence diagrams is infinite under the standard metrics.

Clearly, the opposite change can occur if an edge that should not occur in the filtration is introduced. To apply persistent homology to study data under these constraints, it will be necessary to develop techniques for extracting an approximation of the "real" persistence information. To this end, we are investigating two different approaches:

- Methods of imputing filtration levels where missing edges should be inserted, and
- Methods for constructing complexes for which persistence is stable to deleting edges under some appropriate notion of distance. This report describes the former approach and initial results—the latter is under investigation.

3.3.2.2 Imputing Missing Entries

There are a variety of methods available to impute missing matrix entries. In the case that the matrix is a distance matrix of drawn from a Euclidean point cloud, this problem is well-studied in computational geometry. Similarly, if a matrix is known to be low-rank and the number of missing entries is small, there are a number of methods for finding candidates to replace missing entries. Given the general nature of the problem, and the intended application to non-metric data, it is reasonable to begin with these more general techniques. A standard approach is minimization of the nuclear norm of the matrix, given by the sum of the norms of the eigenvalues. This is a much simpler, robust and more computable criteria than then naive rank minimizaiton, which is carried out by checking that all principal submatrices above a particular size are singular using various combinatorial checks. Like all non-convex optimization problems, the result can be non-deterministic if the problem is badly underspecified, however it is not clear where the threshold for convergence to a correct answer lies, or even if it exists for various classes of data.

3.3.3 Persistent Homology as a Feature Space for Neural Nets

Here, the effort moved to investigate the use of persistence in analyzing the progress of a single fully-connected layer of a convolutional neural network (CNN). Having learned the fundamentals of neural networks, including those with additional convolutional layers, we return to our previous goal of using persistent homology as a tool to augment state-of-the-art machine learning techniques. Specifically, the use of persistent homology on so-called feature space as a technique for engineering convolutional layers of a CNN.

The focus was primarily on developing code and providing a proof of concept for the use of persistent homology in feature space. As we have found in previous investigations persistence appears to be more effective for classification when applied to groups of similar data than individual samples. While this approach is effective at identifying meaningful topological structure within each class it is difficult to apply this structure to unlabeled data, as is the goal for supervised learning tasks. On the other hand, if we were to look at the persistent homology of the features instead of the samples we would be able to identify subsets of features constituting meaningful homological structure within each class that can then be blindly applied to unlabeled data.

This approach was applied to two sets of image data, MNIST and CIFAR-10, both of which contain 10 classes of images. This technique is specifically useful for image data as images generally have a fairly large number of features - pixels, potentially falling into 3 distinct classes ('R', 'G', or 'B', in the case of CIFAR-10) which leaves room for interesting topological structure in feature space. That is, if we were to look at the feature space of 1000 points in 3-dimensional space we would be dealing with 3 points in 1000-dimensional space.

3.3.4 Homological Methods for Sensor Network Integrity

3.3.4.1 Introduction

The purpose of this section is to present research and development progress on questions regarding sensor network integrity using tools from topological data analysis (TDA). The style is intended to be pedagogical, walking the reader through the main ideas, emphasizing pictures over proofs to communicate intuition. The primary work thus far has been to integrate several different tools and combine them with visualization to do experimental data analysis. We will discuss the basic theory and the principal ideas along with examples and their visualization.

Throughout, we will assume that a sensor network is a collection of sensors, deployed in some environment with the ability to detect nearby sensors and do some other types of measurement. The sensors measure some quantity, like temperature, that is assumed to be a continuous function on the domain. In this light, the sensor network is a finite sample from an unknown function on an unknown space. The neighbor information of the sensors is correlated to distances in the domain, but the locations of the sensors (i.e. as coordinates) are not known. It is a challenging theoretical (and practical) setting in which to work. One feels intuitively that there will be very strong limits on what can be computed given the strict limits on the inputs.

However, some very strong theoretical guarantees exist for some fundamental problems in these so-called coordinate-free sensor networks. Perhaps the most immediately compelling of these results is the Topological Coverage Criterion or TCC. This gives a way to extract a guarantee of sensor coverage from the neighborhood information. This result, first formulated by De Silva and Ghrist [5] was extended and simplified by Cavanna et al. [4].

It gives a polynomial-time algorithm to certify coverage. The TCC was a major starting point for the present work, which strives to extend these methods to the analysis of scalar fields and vector fields over sensor networks. That is, we'd like to extract topological information about the unknown function from the network measurements. In particular, we'd like to identify global inconsistencies in the data, which can manifest themselves only in the presence of nontrivial network topology. It is assumed that such inconsistencies are an indication of either sensor errors or possibly intentional manipulation of the network.

Cohomology, and especially persistent cohomology gives a way to identify and localize holes in the data where errors can hide. Interestingly, the theory implies that these are the only places such errors can hide. The basic setting is elaborated in Section 2, where it is explained how the neighborhood graph is augmented to form a larger discrete structure that can be used to represent continuous functions on the the unknown domain. This is the basic object that one computes and visualizes throughout this work. The first topological tool to consider is homology.

In Section 3, we explain the basic principles of homology and how they relate to the complexes of Section 2 and to sensor networks more generally. The main theme is that homology gives a language for describing (and counting) holes in the network in a mathematically rigorous way. Moreover, the holes can be ascribed some "size" or other quantitative information using persistent homology. Section 3 also presents some examples of the visualization of persistent homology in sample networks, showing both the persistent homology (as a persistence diagram) as well as a representative of the most significant hole. Most previous work on homological sensor networks was phrased in terms of homology. However, there is a dual theory of cohomology that is in many senses, the more natural language for expressing and studying functions on the domain. Section 4 gives the basic definitions of cohomology. One interpretation of the cohomology theory we are using is as global structure of a discrete version of differential forms. Seeing as differential forms are a tool for doing caculus without coordinates, it makes sense that discrete differential forms allow us to do some calculus on discrete complexes arising from coordinate-free sensor networks. There is an extensive package by researchers at the University of Illinois implementing Discrete Exterior Calculus [2] that we have integrated into our codebase (See also the book by Grady and Polimeni [7]). We give some examples of using that code to visualize cohomology. Cohomology and the Discrete Exterior calculus have an intimate connection to harmonic analysis. As a result, one can compute the so-called harmonic cocycles. These are used to give embeddings of the network into circular coordinates [6]. Examples and illustrations are given in Section 4. It is expected that these harmonic cocycles can also be used to extend (or interpolate) incomplete data across a network. Last, we report on an implementation of the TCC and show the resulting figures in Section 5.

3.3.5 Multi-Dimensional Scaling of Collections of Persistence Diagrams

3.3.5.1 Introduction

Topological Data Analysis (TDA) is an approach to studying potentially high-dimensional, incomplete, or noisy data by studying its "shape." The primary assumption is that this data carries relevant geometric and topological information about the "system" from which it has been generated. Persistent homology is a popular method in TDA which involves computing a topological signature from point cloud data. Specifically, computing the persistent homology of distance functions involves tracking the scales at which topological features appear and disappear by constructing a nested sequence of simplicial complexes on top of the point cloud at a range of scales. The resulting topological signature known as a persistence diagram, or barcode, is the set of pairs corresponding to the scales at which these features appear and disappear.

Persistent homology has been shown to be a useful tool for studying point clouds that are sampled from a single unknown space in order to learn something about its structure. We will instead consider collections of point clouds sampled from a family of related spaced defined by a so-called parameter space. Our hypothesis is that by using a metric on the space of persistence diagrams known as the bottleneck distance, we can embed the diagrams of this family of spaces in order to learn something about the structure of the underlying parameter space. Specifically, we utilize Multidimensional Scaling (MDS) in order to embed collections of persistence diagrams in low-dimensional space. MDS is a general purpose tool for embedding metric data into Euclidean space. The special case of MDS on Euclidean point sets is equivalent to principle component analysis (PCA). It strives to minimize the mean squared reconstruction error. One can rewrite this optimization as an eigenvector problem, and the eigenvalues give some indication of the quality of the embedding. To generalize from the case of PCA, the eigenvectors tell us the amount of variance in the data explained by the embedding, a measure of how much information was lost in the visualization process.

3.4 Additional Approaches to Structure Detection

3.4.1 Overview

This section also contains less-topologically motivated means for ascertaining structure in data; the methods are based on the spectrum or singular values of the data. The methods include:

- Informativeness
- Scree Plot Analysis
- Randomized Hadamard Multiplication
- Shannon Entropy

In order to facilitate the subsequent discussion the singular value decomposition (SVD) of an NxN matrix *D* will be defined as:

$$D = \left[U \Sigma V^T \right], \tag{3}$$

where *U* is a matrix whose columns consist of the left-singular vectors, Σ is a diagonal matrix consisting of the singular values σ_i, where $i \in \{1, ..., N\}$ and V^T is a matrix whose columns consist of the right-singular vectors [16].

3.4.2 Informativeness

The work of [2] introduces a new framework for measuring heterogeneity in correlation matrices. They defined the *informativeness* of a correlation matrix as the distance be-tween a given correlation matrix and the nearest non-informative correlation matrix. A non-informative correlation is defined as one with constant off-diagonal entries. In par-ticular, [2] expands and generalizes the techniques by [17], [18] and [19] for comparing correlation matrices. This approach allows the application of a wide range of distance measures that can be applied to measuring the difference between a correlation matrix and a non-informative correlation matrix.

A correlation matrix A is an SPD NxN real matrix, such that $A_{i,i} = 1$, i.e. is normalized so that each diagonal element is equal to a constant value a. In [2], various measures were used to estimate the homogeneity of a correlation matrix; in addition to classical techniques such as Bartlett's or Lawley's methods, newer norms such as the Bures distance were also studied and simulated. Following [2], a non-informative (correlation) matrix is generally a SPD matrix whose off-diagonal elements are all equal to 1. Specifically, the set of non-informative matrices is given by a convex combination of a constant matrix J and centering matrix H [2] as

$$\mathcal{N} \equiv \left\{ N_a = aJ + (1-a)\frac{n}{n-1}H : 0 \leq a \leq 1 \right\}, \tag{4}$$

where $H = I - \frac{1}{n}J$ and $J = \mathbf{1}\mathbf{1}^T$.

Given the above definition, the informativeness of a correlation matrix C is defined to be the distance from the correlation matrix to the nearest non-informative matrix N_a [2]

$$\begin{aligned} d_{\mathit{N}}(C) &\equiv \min_{N \in \mathcal{N}} d(C, N) \\ &= \min_{0 \leq a \leq 1} d(C, N_a). \end{aligned} \tag{5}$$

Note that in contrast to the Riemannian distances between two covariance matrices, the informativeness is a single-sample measure which operates on a single correlation matrix. In this work, two distance measures are explored for d in Eq. (5). The first distance measure is the Euclidean distance d_F

$$d_{EUC}(C_1, C_2) = \|C_1 - C_2\|_F\,, \tag{6}$$

based on the Frobenius norm, which is given by the square root of the sum of squared matrix entries. E.g. the Frobenius norm of an NxN matrix C with real entries is

$$\|C\|_F = \sqrt{\sum_{i,j=1}^{N} c_{i,j}^2}. \tag{7}$$

Another distance measure explored in [2] is the Bures distance d_B

$$d_B(C_1, C_2) = \sqrt{2 - 2\left\|\sqrt{C_1}\sqrt{C_2}\right\|_*}, \tag{8}$$

where $||\cdot||_*$ is the nuclear norm of a matrix.

3.4.3 Scree Plot Analysis

A plot of the sorted singular values σ_i, where $i \in \{1, ..., N\}$ on a graph in descending order is called a *scree plot* [3]. An attempt is then made to determine the index corresponding to the singular value where the graph appears to flatten or level out. A principled way of doing this is to perform numerical differentiation on the set of singular values. Then the maximum difference indicates the largest drop in singular values, corresponding to a suitable cutoff point, beyond which singular values are not significant. Determining this threshold at which to cutoff is key. There are two methods for scree plot analysis in [3]: direct computation and maximum likelihood estimation.

3.4.4 Random Hadamard Multiplication

Another method to detecting structure begins by multiplying the SPD matrix by a random matrix and is based on [3] and [20]. The random sensing matrix Φ must possess cer-tain properties, such as independence, a zero mean, and a low variance (similar to the properties typically required for sensing matrices to possess a low coherence in the field of *compressive sensing*). The premise is that if the data matrix D does not contain a significant amount of structure, then element-by-element multiplication by a random binary $+1/-1$ matrix will not alter the 2-norm of the data matrix, i.e., $\|D\|_2 \approx \|\Phi D\|_2$ [3]. If there exists structure, then $\|D - \Phi D\|_2$ will be commensurate with the amount of structure present. Normalizing by the Frobenius norm of the data matrix gives

$$h = \frac{\|D\|_2 - \|\Phi D\|_2}{\|D\|_F}, \tag{9}$$

where Φ is a random matrix whose i^{th}, j^{th} entry is defined as $\phi_{ij} \in \{+1, -1\}$.

3.4.5 Shannon Entropy of Singular Values

In order to determine the significant subspaces of a data set, the Shannon entropy of the singular values of the data set can be computed [3]. First, the square of each singular value is normalized by the sum of all singular values, as

$$f_k = \frac{{\sigma_k}^2}{\sum_{i=1}^{N} \sigma_i^2}, \tag{10}$$

where σ_i and σ_k are the i^{th} and k^{th} singular values, respectively for $i, k \in \{1, ..., N\}$. Then, the entropy of the set of singular values is computed as

$$H(f) = \frac{-1}{log(N)} \sum_{k=1}^{N} f_k log(f_k), \tag{11}$$

where $log(N)$ is the natural logarithm of N.

3.4.6 Distributions of Matrix Entries

In [21], the distribution of off-diagonal elements of the normalized Gram (correlation) ma-trix was introduced and analyzed, in order to assess geometric changes in overcomplete dictionaries. Doing so corresponds to the distribution of the data vectors on the surface of the unit hypersphere, a measure of sphericity of the data. The Kolmogorov-Smirnov and Kullback-Leibler divergence was used to measure distances between the distributions of off-diagonal correlations to ascertain the degree of dictionary change.

As an alternative to comparing the mean of Gram matrix entries, the distance between Gram matrix entry distributions provides more detailed geometric information about the changes in the effective dictionary due to dictionary updates. It is suggested that traditional statistical distances can be applied to a distribution of Gram matrix entries to detect a shift in effective dictionary geometry. Statistical distances can be used to monitor the difference between the distributions of entries of the Gram matrix of the optimized CS system and Gram matrix entries of the stale CS system, seeking a change in the geometry of the effective dictionary. These statistical distances (presuming that their computational complexity is low enough relative to re-generating the sensing matrix) could be used to alert the system that an updated sensing matrix should be generated.

To compare Gram matrices distributions, traditional methods of probability distribution distances can be used, of which there are many [22]. Two well-known measures to compare distributions are the *Kolmogorov-Smirnov* test and the *Kullback-Leibler* divergence.

3.4.6.1 Overview of Statistical Distances

Reference [23] identifies two significant applications of statistical distances in data analysis:

- Parametric statistical inference (estimating parameters of a presumed distribution)
- Goodness of fit (measuring distances between a data set and a known distribution)

Many statistical distances are used for both purposes, and there are generally two types of distances:

- Distances between distribution functions, e.g. Kolmorgorov-Smirnov, Cramer-Von Mises, Anderson-Darling
- Distances between density functions, e.g. Kullback-Leibler Divergence, Bregman Divergences, Pearson's Chi-Square, Hellinger Distance

This effort examines one of the distribution distances from [21], namely the Kolmorgorov-Smirnov (KS) statistic.

3.4.6.2 Kolmogorov-Smirnov Test

The KS test is a non-parametric statistical *goodness of fit* test that was originally developed to determine if an empirical data sample set belongs to a given continuous underlying CDF [22]. Smirnov later extended the KS-test to compare two empirical data distributions, i.e. EDFs [22]. To begin, define $X_1, X_2, ...X_m$ and $Y_1, Y_2, ...Y_n$ as two independent samples from populations with unknown EDFs, defined as F_X and F_Y. In addition, define F_m and F_n as the corresponding empirical EDFs. Then, the hypothesis test for the equivalence of the two distributions is formulated as [22]:

$$\begin{aligned} H_o &: F_X(x) = F_Y(x)\ \forall x \\ H_1 &: F_X(x) \neq F_Y(x)\ \exists x \end{aligned} \tag{12}$$

The KS-statistic for comparing two distributions is [22]:

$$D_{KS}(m,n) = \underset{x}{sup}\,|F_m(x) - F_n(x)| \tag{13}$$

The KS-statistic translates to the maximum difference between the distribution function of $F_m(X)$ and the distribution function of $F_n(x)$.

There is a significant body of literature that applies the KS-statistic to perform a binary hypothesis test as to whether two data sets are drawn from the same distribution. In addition to this, however, the *KS-statistic* $D_{KS}(m,n)$ can be used as a rough measure of distance between two data sets without performing a statistical test, *per se*. The KS-statistic can provide a relative measure, in some sense, of how *far* one data set is from another data set.

3.4.6.3 Applying the KS Test to SPD Matrix Evaluation

In this section, the KS statistic is applied as a distance measure between the entries of a SPD matrix $G(A_{t1}, A_{t2})$ and an identity matrix of the same size. Define $G_{t_2}(:)$ to be the discrete finite data set consisting of the absolute value of the entries that lie above the main diagonal of the *updated* SPD matrix. Similarly, define $G_{t_1,t_2}(:)$ to be the discrete finite data set consisting of the absolute value of the entries that lie above the main diagonal of the *stale* SPD matrix. The total number of elements in either matrix is MxM, so the total number of elements above the main diagonal of either SPD matrix is

$$N_{triu} = \frac{M\,(M-1)}{2}. \tag{14}$$

Therefore, each of the data sets $G_{t1}(:)$ and $G_{t1,t2}(:)$ contains N_{triu} elements of real scalars taking values on $[0,1]$. Define $F_{t1,t2}(x)$ to be the EDF of the SPD matrix $G_{t1,t2}(:)$. The $N_{t1,t2}$ points used to evaluate the EDF are within the interval $[min(G_{t1,t2}(:)), max(Gt1, t2(:))]$. This interval is sub-divided into $N_{t1,t2}$ sub-intervals, whose bin edges are collected in the $N_{t1,t2}$-element vector $x_{t1,t2}$. Then the EDF $F_{t1,t2}(x)$ is given by

$$F_{t1,t2}(x) = \frac{1}{N_{t1,t2}} \sum_{i=1}^{N_{t1,t2}} \mathtt{I}(G_{t1,t2}(:) \leq x_{t1t2}(i)). \tag{15}$$

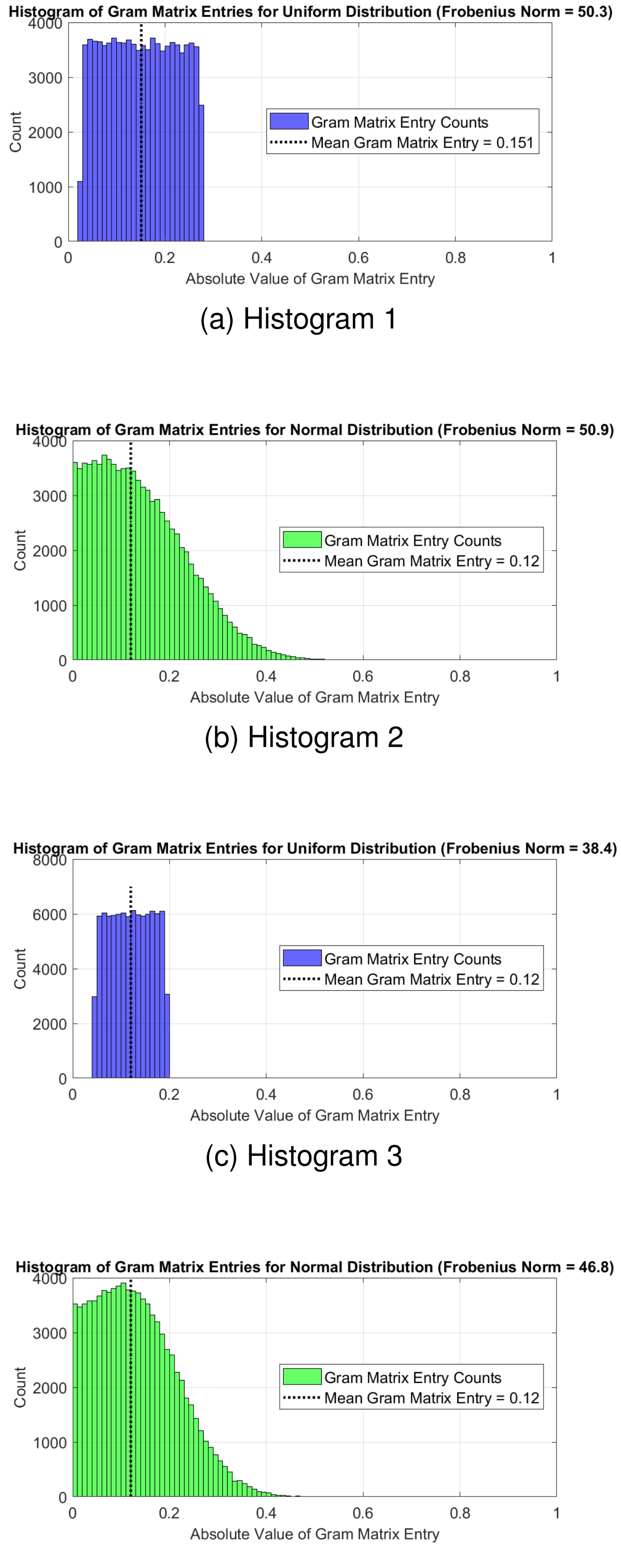
The KS statistic then acts as a difference operator between the two SPD matrices

$$\mathcal{D}_{KS}\left\{G(A_{t_1,t_2}), I_N)\right\} = \underset{x}{sup}\, |F_{t_1,t_2}(x) - F_{t_1}(x)| \tag{16}$$

where the argument x indexes the EDFs over the interval $(-\infty, \infty)$. The N_{bin}-element vector x represents the N_{bin} bin edges into which the interval is divided. The vector x can be expressed in the MATLAB language as $[-\infty; sort([G(A_{t_1,t_2}(:); G(A_{t_2}(:)]); \infty]$. The associated KS-statistic offers some interesting geometric interpretation. One interpretation of the KS-statistic in light of the geometry of the EDF of Gram matrix entries is that the KS-statistic is the largest difference between two dictionaries, regarding the number of atom pairs that are at least a given distance apart from each other.

In order to provide some motivation for seeking a measure beyond the Frobenius norm, an example using a contrived data set to simulate the entries of the Gram matrix is presented in Fig. 4. Data sets (representing Gram matrix entries) are constructed to demonstrate that Gram matrices with very similar Frobenius norms, or with very similar means of their entries can have very different histogram distributions. The different histogram distributions represent significantly different underlying effective dictionary geometries. Therefore, single-parameter measures of the Gram matrix (e.g. only the mean

Gram matrix entry, or only the Frobenius norm of the Gram matrix), may in some cases not provide a good measure of the relative geometry of the data vectors. Consequently, other Gram comparison methods are sought which more comprehensively encode the geometry of the data.



(a) Histogram 1

(b) Histogram 2

(c) Histogram 3

(d) Histogram 4

Figure 4: Failure of Single Parameter Gram Matrix Measures. Histograms of synthetically generated Gram matrices are shown. Neither the Frobenius norm nor the mean Gram matrix entry reflect the geometry of the histograms, and therefore the geometry of the underlying data sets.

3.4.7 SPD Matrix Difference Measures

The methods explored previously use the persistent homology of a correlation or Gram matrix. In contrast, the tools studied in this section offer less topological methods for finding interesting geometric features or changes in the geometry of data. One way to as-certain geometric change in data sets is to look at the data sets through the lense of their respective covariance matrices, in essence representing each data set by its covariance matrix. Then, the space of SPD matrices, each covariance and therefore each data set becomes a point and distances can be computed between and among data sets. In this section, four categories of SPD matrix measures are described: Euclidean matrix mea-sures, metrics between distributions of off-diagonal matrix elements, informativeness (or homogeneity) and Riemannian distance metrics.

This are was briefly studied (but mainly left to a future research effort) was distances between pairs of SPD matrices, which can include Gram matrices, correlation matrices and Riemannian distances. In a parallel context, the informativeness concept applies distances between the test affinity matrix and the nearest non-informative matrix. However the Riemannian distances that were tested in [2] were rejected due to the issue of infinite distances being returned [2]. Furthermore, as a means for computing a distance *between two* matrices, the structure found within an affinity, correlation, Gram, or covariance can be characterized by its distance from the non-informative identity matrix, as is done in the results in Section 4.7.5.

3.4.7.1 Euclidean Distance Measure

The Frobenius norm of a matrix is the matrix equivalent of taking the Euclidean norm of a vector. It vectorizes the matrix argument and disregards any structure inherent in the matrix. The Frobenius norm of a matrix C is

$$\begin{aligned} \|C\|^2_{Fro} &= \sqrt{tr(C^T C)} \\ &= \sqrt{\sum_{i,j=1}^{N} c^2_{i,j}}. \end{aligned} \tag{17}$$

Consequently, perhaps the most straightforward way to compute a distance between two SPD matrices C_1, C_2 is to take the Frobenius norm of the difference of the two matrices, given by

$$d_{EUC}(C_1, C_2) = \|C_1 - C_2\|_{Fro} \tag{18}$$

3.4.7.2 Distances on the Riemannian Manifold

The affine-invariant Riemannian metric (AIRM) [9] is a metric applied to the space of MxM symmetric positive definite (SPD) matrices. The distance between NxN covariance matrices C_1 and C_2 that follows from the AIRM is [24], [9]

$$\begin{aligned} d_{aie}(C_1, C_2) &= \left\| log\left(C_1^{-1/2} C_2 C_1^{-1/2}\right)\right\|_F \\ &= \sqrt{\sum_i^N log^2\left(\lambda_i(C_1^{-1} C_2)\right)}, \end{aligned} \tag{19}$$

where λ_i is the i^{th} eigenvalue of the product $C_1^{-1}C_2$ and log refers to the matrix logarithm.

3.4.8 Impact of Record Length on Covariance Measures

When using any of the types of SPD matrix measures that are mentioned in this report, the number of data records accumulated prior to computing a particular affinity matrix (e.g. covariance, correlation, Gram, etc.) will impact the affinity matrix structure itself, as well as distances such as the AIRM or correlation homogeneity measures such as informativeness [15]. Even if each of two data sets are drawn from the same distribution, the Euclidean geodesic distance between the two respective covariance matrices will vary greatly when the number of records from one data set is vastly different from the other. Furthermore, both theory and simulation shows that the covariance distance will even be significant between a data set and a subset of the same data set.

As an example, assume that a collection of measured data vectors x_j is distributed according to a multi-variate normal distribution, and each data vector is stored in a ma-trix $X_j = [x_{1j}, ..., x_{nj}] \sim \mathcal{N}(0, \Sigma)$, with $j = 1, ..., L$. Then the expected value of $d_{fro}(k)$ for $k \in [2, ..., T-1]$ is given by

$$E\left[d_{fro}(k)\right] = \left(\frac{1}{k} + \frac{1}{L-k}\right) \left(tr\left(\Sigma^2\right) + tr\left(\Sigma\right)^2\right), \tag{20}$$

where $d_{fro}(k) = \|C(1,k) - C(k+1, L)\|_F^2$, and $C(i,j) = \frac{1}{j-i+1}\sum_{\ell=i}^{j}\left(X_\ell - \mu_X\right)\left(X_\ell - \mu_X\right)^T$, L is the entire record data length, k is the number of records used to compute the first covariance matrix $C(1,k)$ and $L-k+1$ is the number of records used to compute the second data matrix $C(k+1, L)$. The parameter index k indicates the time or sample at which one data record length ends and the next begins.

The proof of this equation is provided in the appendix in Section 6, and is based on the outline given in [15]. This effect is present in all of the SPD matrix measures and in practice needs to be identified and accounted for when utilizing any type of matrix difference as applied to affinity matrices such as covariance, correlation, or Gram.

3.5 Related Tools Identified for Future Study

In addition, the following list of tools to perform structure detection were found in the literature, but were not explored in this research [3]:

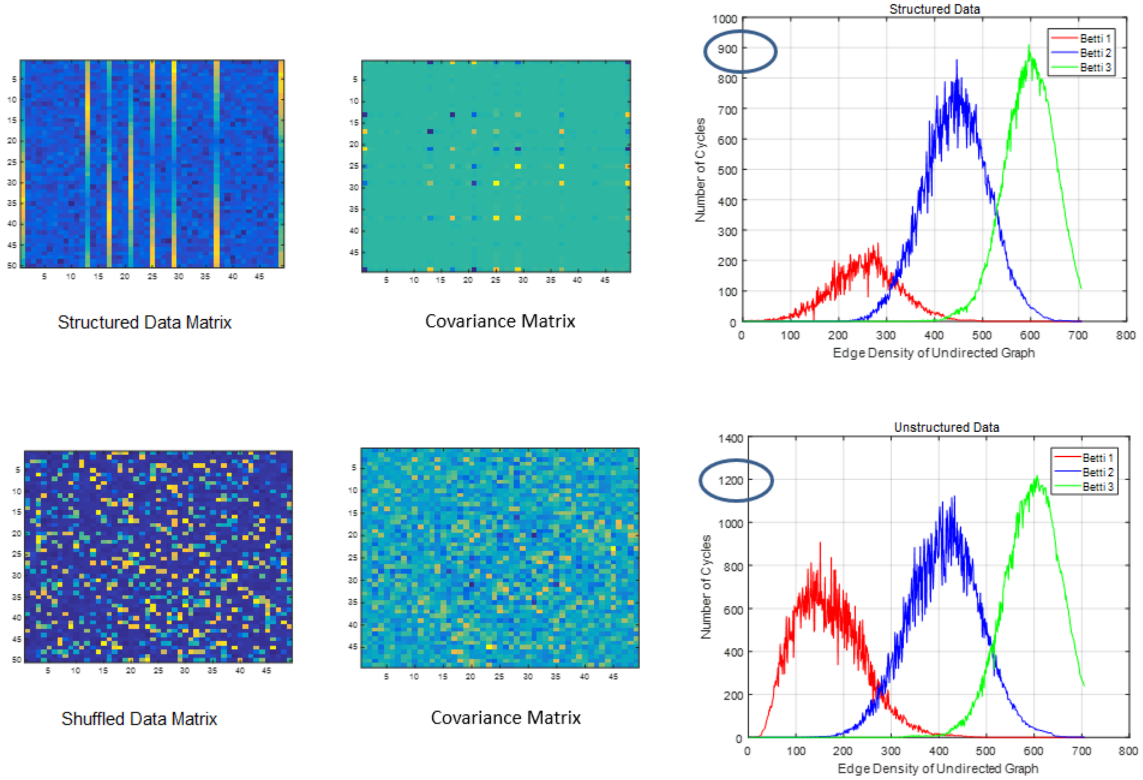
- Independent Component Analysis
- Semi-Discrete Decomposition
- Non-Negative Matrix Factorization
- Tensor Analysis

4 Results

4.1 Structure Detection Using Betti Curve Area

Fig. 5 shows the data flow for a single run of this algorithm. Note that the shuffling of the data matrix has a significant impact on the appearance of the covariance matrix, as compared to the structured case. The covariance matrix that was spikey in the structured case now becomes much more dense than in the structured case.

Running this algorithm many times in a Monte Carlo type experiment results in a distribution curve for each of the Betti numbers, or what is referred to as a *Betti curve*. The Betti curves generated from multiple Monte Carlo runs are shown in the lower right corner of Fig. 6. Lastly, in the lower left corner of Fig. 6, the sum of the areas of each respective Betti curve is shown. These sums provide an aggregate signature of the underlying topological characteristics of the data over all of the Monte Carlo runs.


Structured Data Matrix

Covariance Matrix

Shuffled Data Matrix

Covariance Matrix

Figure 5: Initial Prototype Example-Single Run

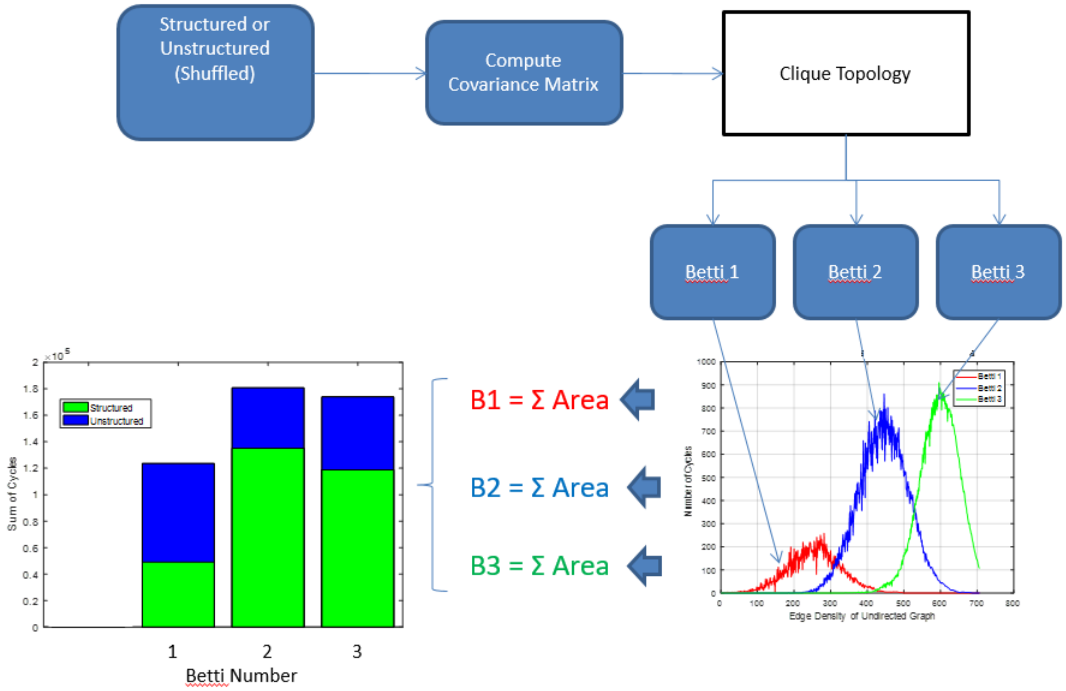


Figure 6: Data Flow for Initial Prototype Example

A baseline experiment using Betti curve areas was performed using 8 correlated subspsaces. The corresponding bar chart is shown in Fig. 7. Note that in the experimental results, the structured case contains in general fewer cycles for all three Betti numbers than the shuffled or unstructured case. It is hypothesized that the randomness of the unstructured shuffled data allows for more connectivity than in the structured case, and therefore the unstructured data results in a higher Betti number count and therefore a greater area under the corresponding Betti curves.

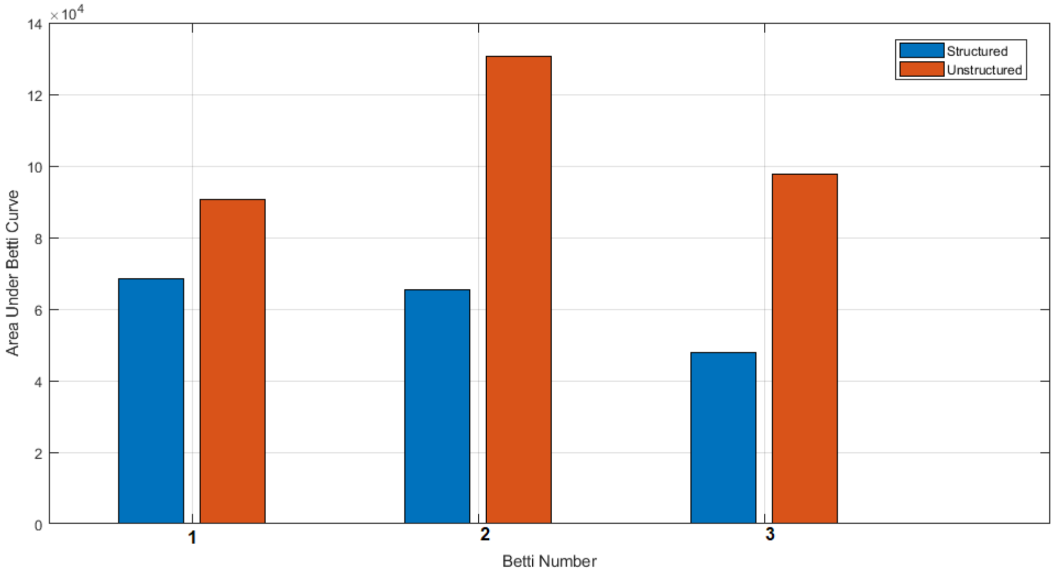


Figure 7: Betti Curve Areas (Eight Correlated Subspaces)

A subsequent additional experiment was conducted where the number of correlated components was decreased from 8 to 4. The resulting bar graphs of the Betti curve areas are shown in Fig. 8. When comparing Fig. 8 to Fig. 7, it can be seen that the decrease in correlated components from 8 down to 4 significantly narrows the gaps between the corresponding Betti curve areas, for all three Betti numbers. In addition, decreasing the number of correlated components from 8 to 4 evidently causes the area under Betti curve 3 to exhibit a reversed relationship. I.e., in Fig. 7, Betti curve area number 3 was higher for the unstructured case when compared to the structured case; for 4 correlated components, the unstructured case has a lower Betti curve area than the structured case, as seen in Fig. 8.

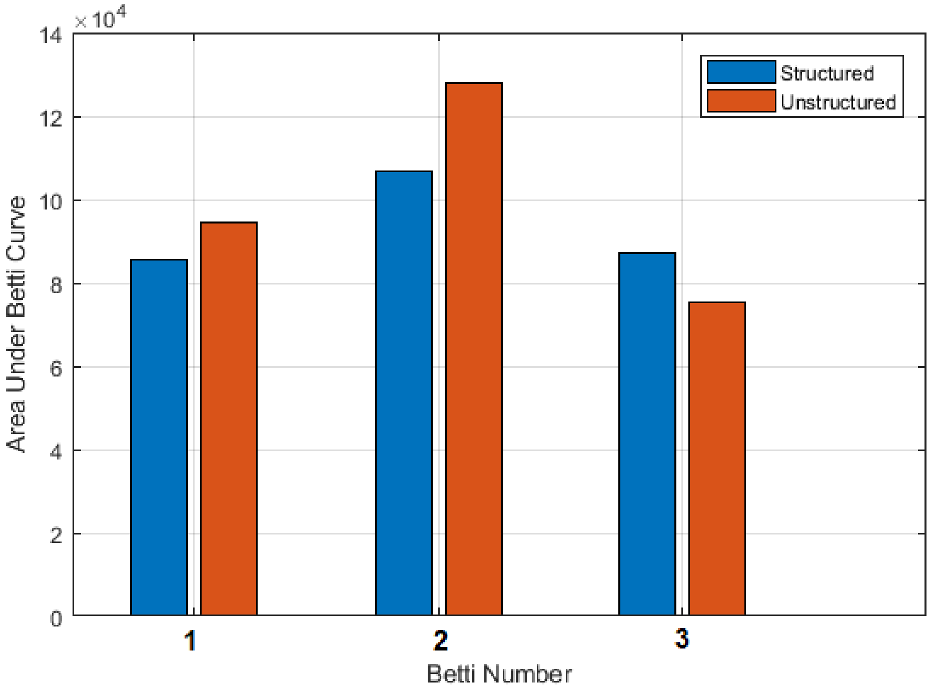


Figure 8: Betti Curve Areas (Four Correlated Subspaces)

4.2 Persistence with Incomplete Pairwise Measurements

As part of our initial investigation into the appropriateness of the method, we have run a series of numerical experiments. Beginning by selecting points in Euclidean space us-ing a uniform distribution on the unit cube. We measure similarity of the vectors using a linear transformation of the standard metric, covariance and absolute correlation. We then threshold the matrix, discarding a sequence of percentages of the lowest similarity edges (modeling sparsification) or a random sample of the data (modeling missing data), and run nuclear norm minimization to recover a low rank matrix which explains the obser-vations. (Note that we do not necessarily expect even the original matrices to be low rank.) Further experiments are planned using other starting data, including data gathered from the Stanford Network Analysis Project (SNAP). As expected, in each case the com-

pleted matrices converge entry-wise to the original as the number of deleted entries shrinks, and so we see a convergence in the diagrams appear to converge to the diagram of the base matrix. As might be expected, the rate of convergence – that is, the relationship between the number of missing entries and the success of the optimization in approximating the matrix – seems to vary substantially between similarity measures. There is a corresponding convergence in the persistence diagrams, which we are currently seeking to understand through the lens of the MDS embedding described in the the accompanying white paper, "Multidimensional Scaling of Collections of Persistence Diagrams". While we are still analyzing the underlying phenomena and structure, this approach appears promising in the case of matrices with relatively small numbers of missing entries.

However, this approach is difficult to understand due to the stochastic nature of the method, and thus it is correspondingly difficult to produce analytic bounds on the error. Therefore the focus ought to shift to investigating other, more analytically tractable methods of imputing the missing entries. For example, one could search for places to introduce edges which minimize the total persistence (using the intuition that missing edges create cycles with long lifetimes and that most real systems have relatively small total persistence), as well as modern techniques drawn from the study of network flows (searching for the weight to assign a missing edge which causes the least change in the spectrum of the Laplacian). It is intuitive that these two approaches should be related, and both offer substantial advantages in terms of formal mathematical analysis.

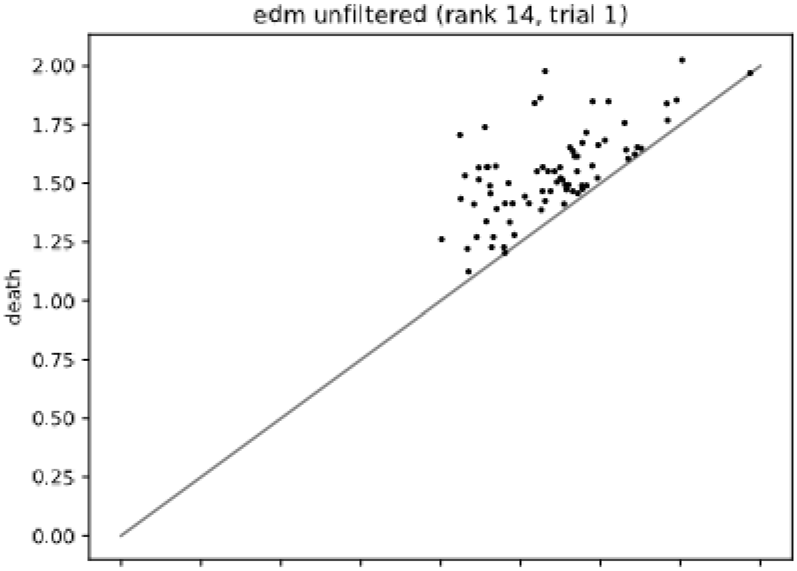


Figure 9: Control Persistance Diagram

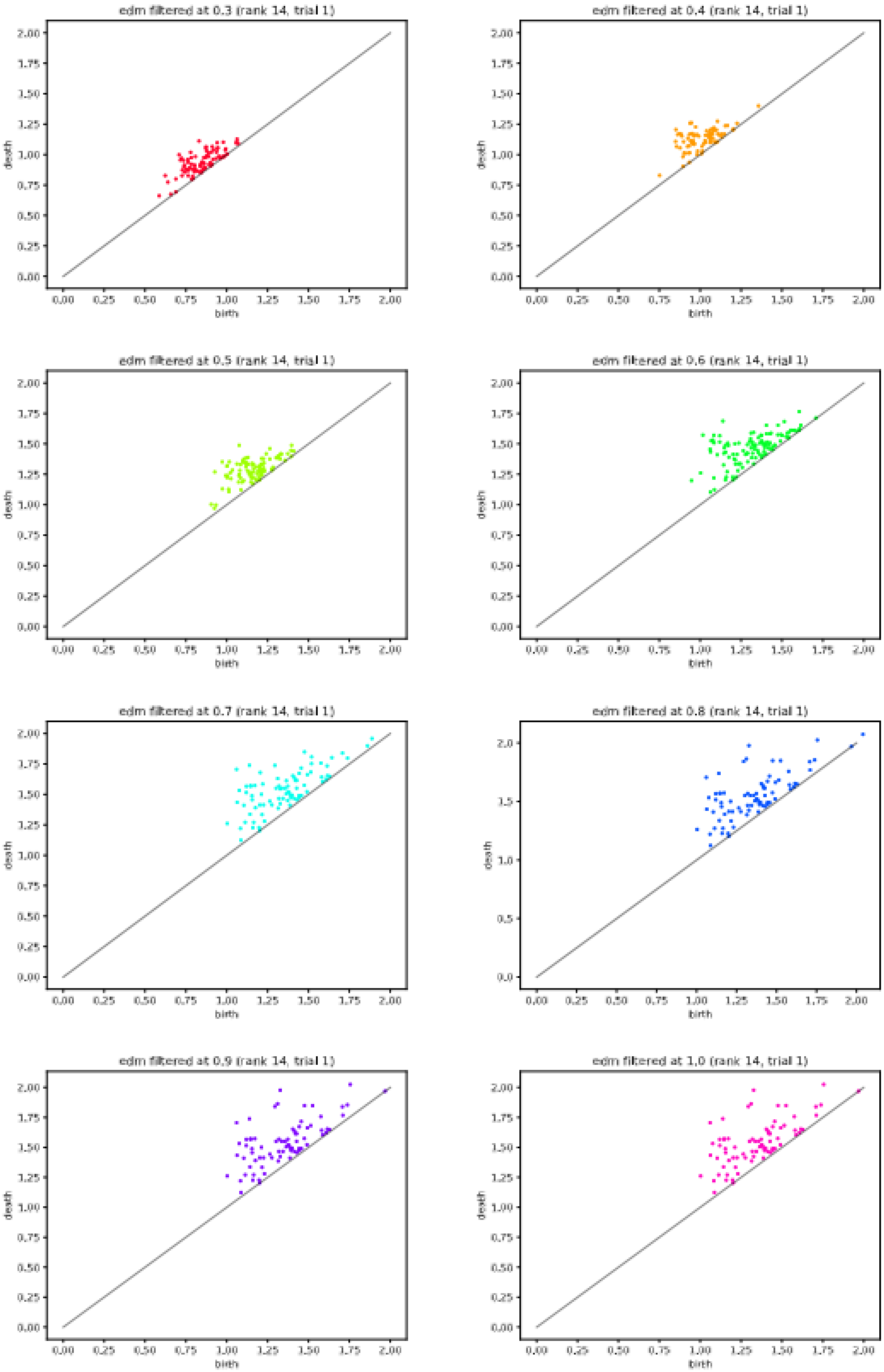


Figure 10: Persistence Diagrams of Thresholded EDMs

4.3 Persistent Homology on Feature Space for Neural Nets

We first applied the technique to the MNIST dataset consisting of 60,000 training and 10,000 testing samples, each 28x28 grayscale images falling into 10 classes - handwritten digits 0-9. That is, our training data can be broken up into 10 groups of 6,000 samples in of each digit in 784-dimensional space. In feature space, we have 10 groups of 784 points in 6,000 dimensional space. We calculated the 1-dimensional persistent homology of each of these groups and found the representative cycles of each point in each the persistence diagram - topologically significant cycles of edges between pixels, which we simply interpreted as subsets of pixels. Fig.11 shows the sum of these subsets as masks weighted by their persistence (death - birth) - that is, for each digit, we plotted the sum of 28x28 grids, one per persistent feature, in which a pixel is zero if it is not present in the representative cycle of a feature, and death - birth if the cycle is present in the cycle of a feature.

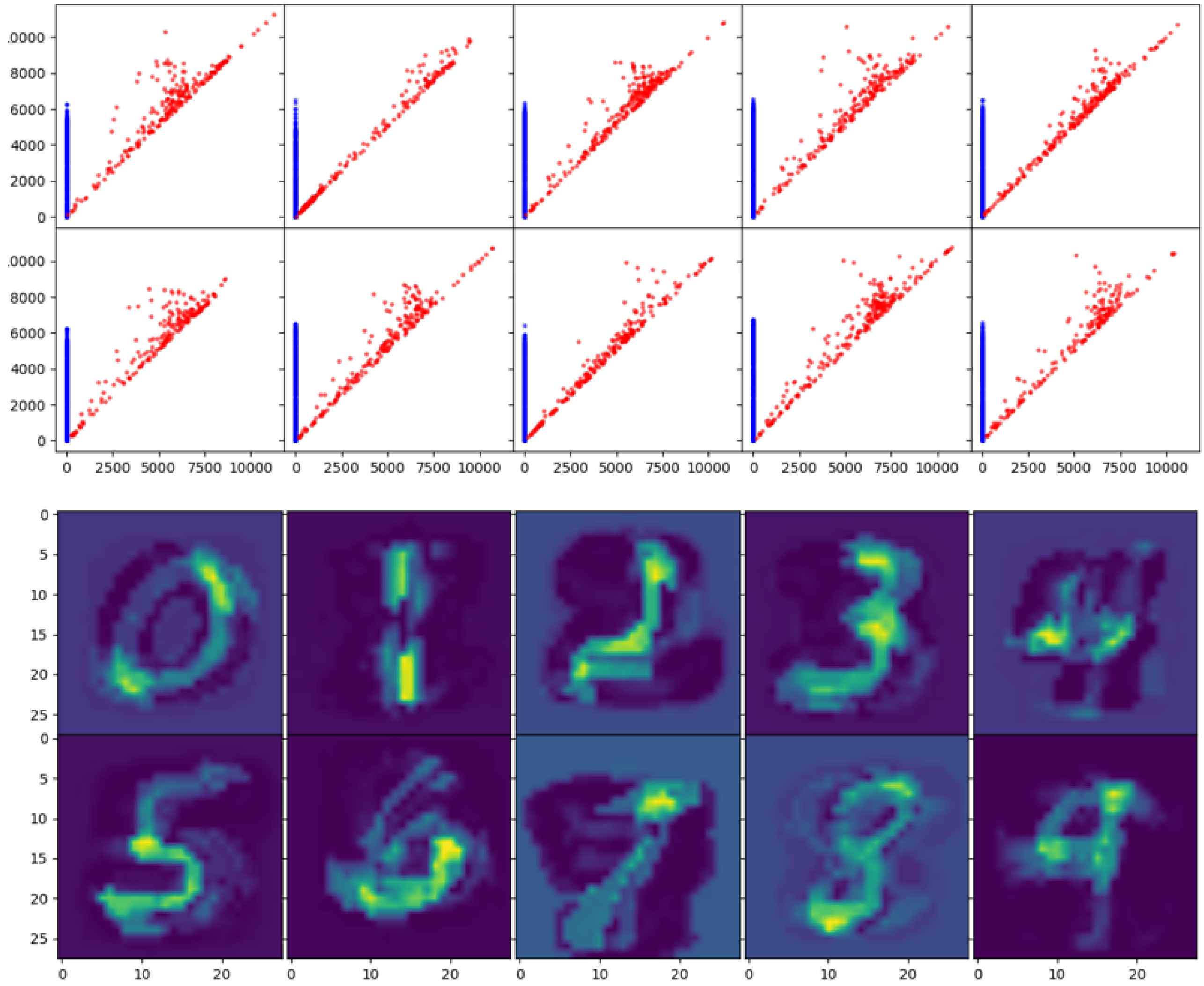


Figure 11: Persistence Diagrams of Handwritten Digits

We applied the same technique to the CIFAR-10 dataset consisting of 50,000 training and 10,000 testing samples, each 32x32 color (RGB) falling into 10 classes such as ship, truck, cat, dog etc. We chose this dataset for its irregularity - unlike MNIST the objects of interest are not “centered” in the plane. Moreover, as we are now dealing with color images the code developed for MNIST was modified to account for multiple “channels.” Instead of constructing one persistence diagram per class we constructed three per class, one for each channel. Fig. 12 shows the resulting masks in which the three masks constructed independently are combined for each class to form RGB image tensors.

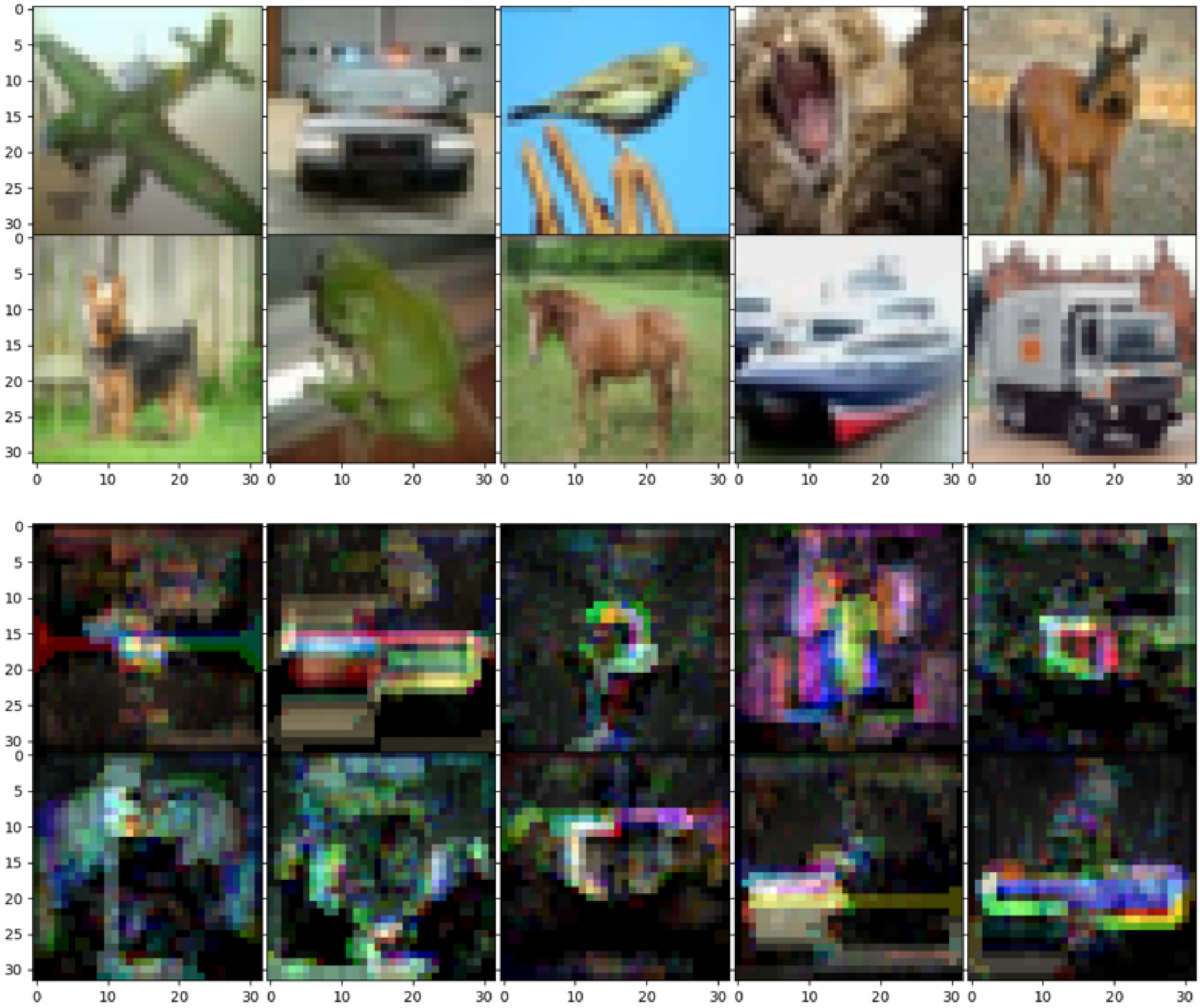
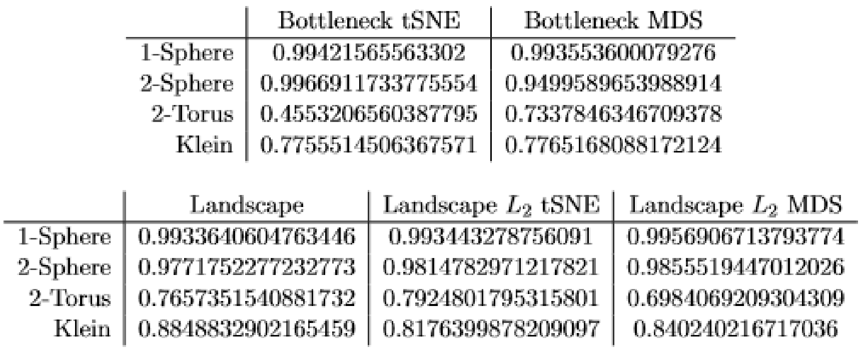


Figure 12: Masks Developed for Images

4.4 Parameter Space Reconstruction from Embeddings of Persistence Diagrams

The following results are adapted from Dr. Sheehy's and Dr. Giusti's work.



	Bottleneck tSNE	Bottleneck MDS
1-Sphere	0.99421565563302	0.993553600079276
2-Sphere	0.9966911733775554	0.9499589653988914
2-Torus	0.4553206560387795	0.7337846346709378
Klein	0.7755514506367571	0.7765168088172124

	Landscape	Landscape L_2 tSNE	Landscape L_2 MDS
1-Sphere	0.9933640604763446	0.993443278756091	0.9956906713793774
2-Sphere	0.9771752277232773	0.9814782971217821	0.9855519447012026
2-Torus	0.7657351540881732	0.7924801795315801	0.6984069209304309
Klein	0.8848832902165459	0.8176399878209097	0.840240216717036

Figure 13, Bottleneck Distances

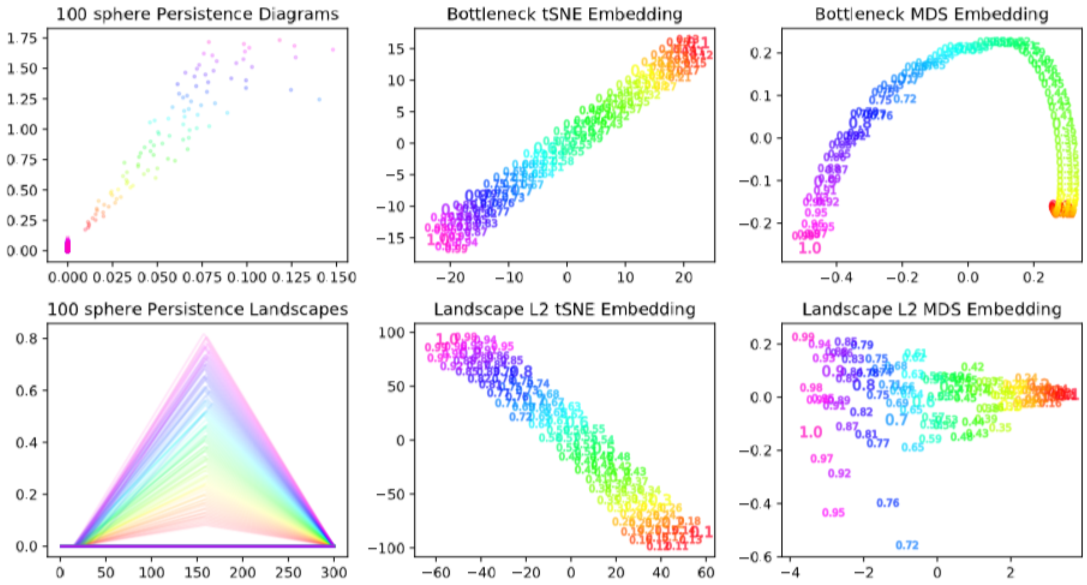


Figure 2: Embeddings of 100 distinct 1-sphere samples generated from uniformly spaced parameters $r \in [0.1, 1.0]$ corresponding to the radius of the spheres.

Figure 14, Embeddings of 1-Sphere

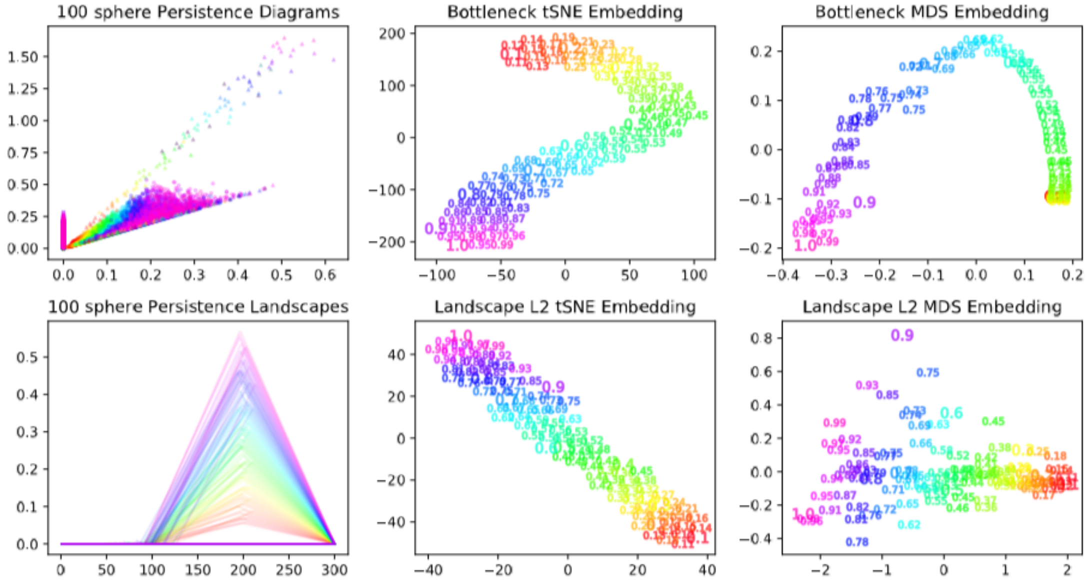


Figure 3: Embeddings of 100 distinct 2-sphere samples generated from uniformly spaced parameters $r \in [0.1, 1.0]$ corresponding to the radius of the spheres.

Figure 15, Embeddings of 2-Sphere

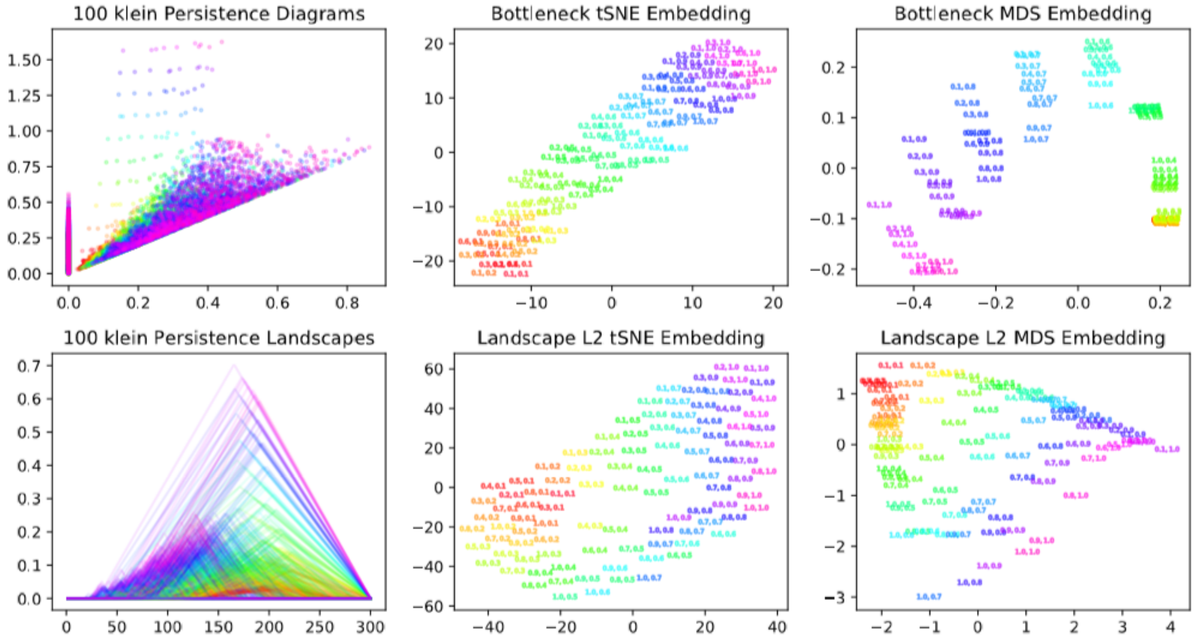


Figure 4: Embeddings of 100 distinct klein bottle samples generated from uniformly spaced parameters $(r, R) \in [0.1, 1.0]^2$.

Figure 16, Klein Bottle Embeddings

4.5 Multi-Dimensional Scaling of Collections of Persistence Diagrams

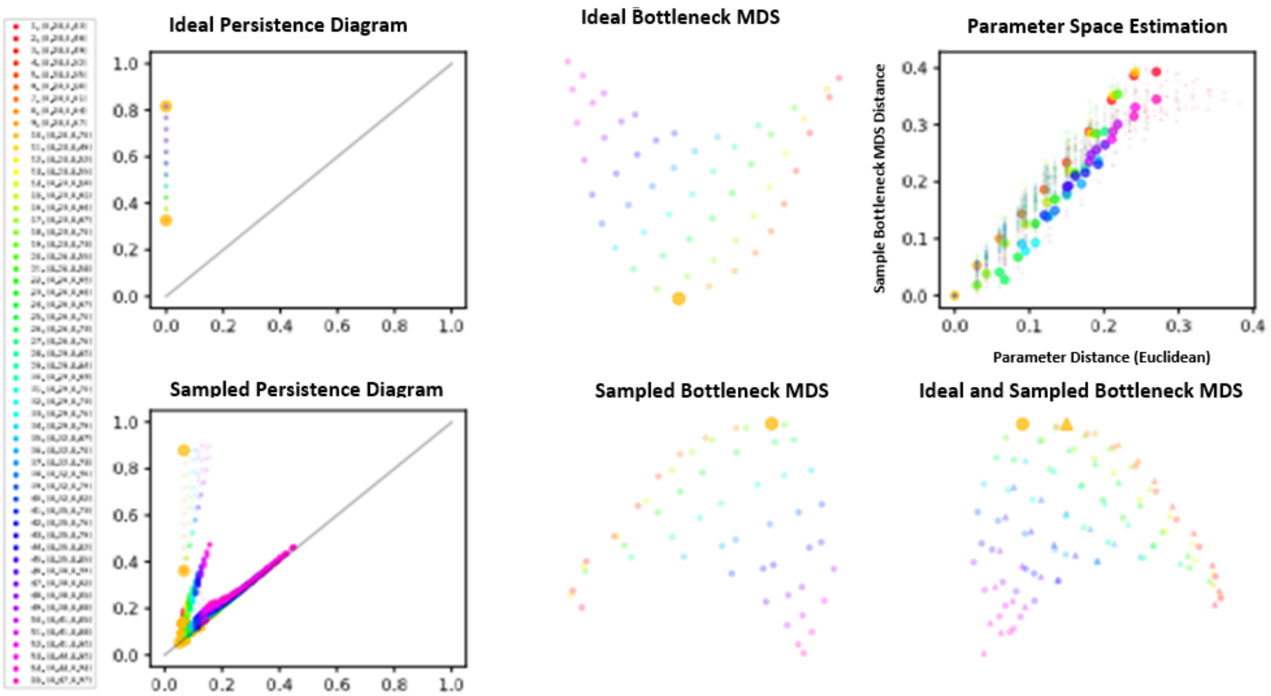


Figure 6: Plot of the pair $p_{10} = (0.2, 0.7)$ (aspect ratio ∼ 3.5).

Figure 17, MDS of Persistence Diagrams (p_1,0)

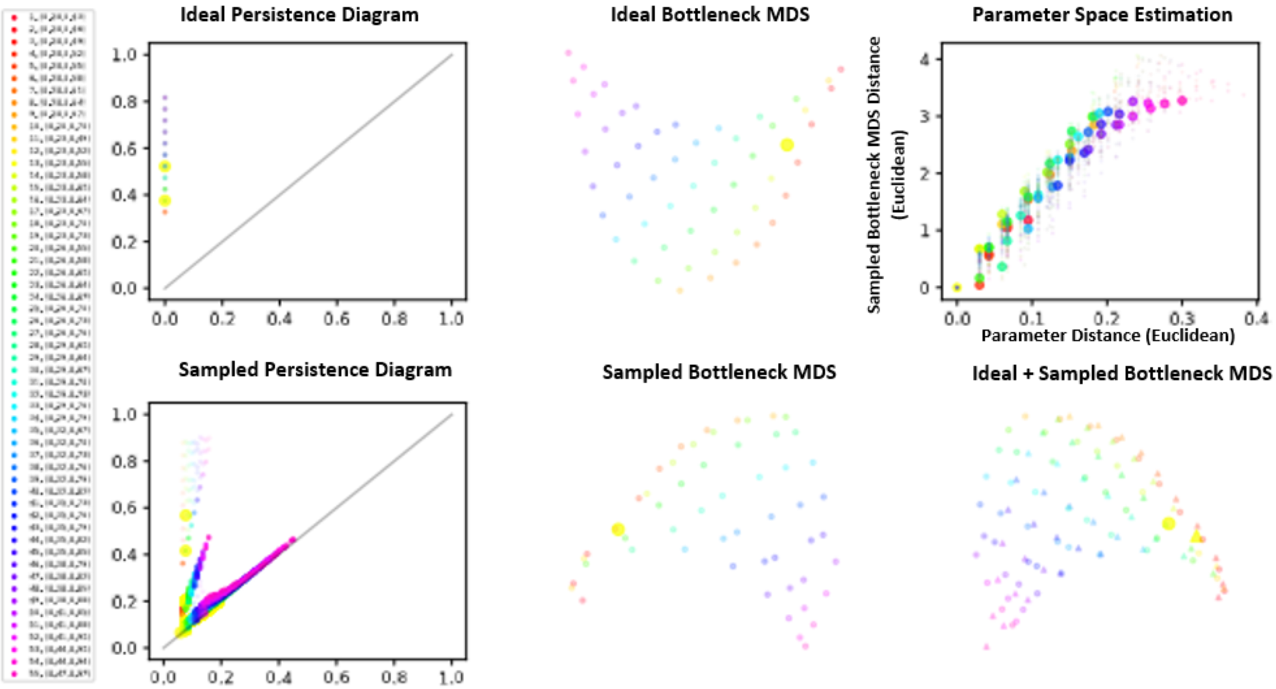


Figure 7: Plot of the pair $p_{13} = (0.23, 0.55)$ (aspect ratio ~ 2.39).

Figure 18 MDS of Persistence Diagrams (p_1,3)

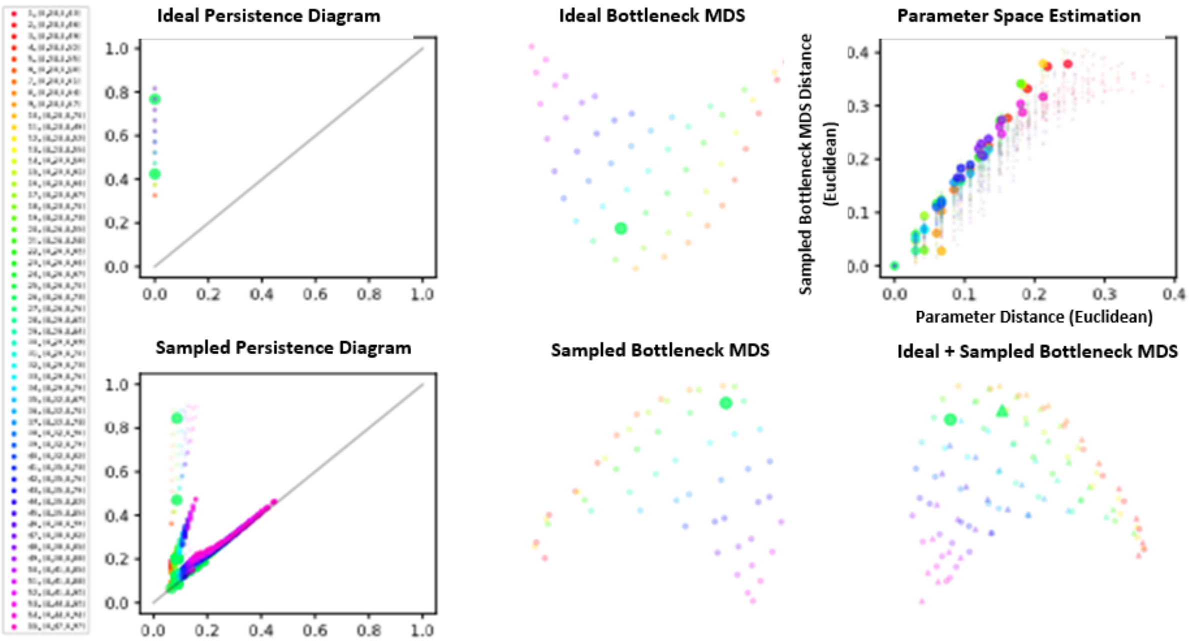


Figure 8: Plot of the pair $p_{26} = (0.26, 0.73)$ (aspect ratio ~ 2.81).

Figure 19 MDS of Persistence Diagrams (p_2,6)

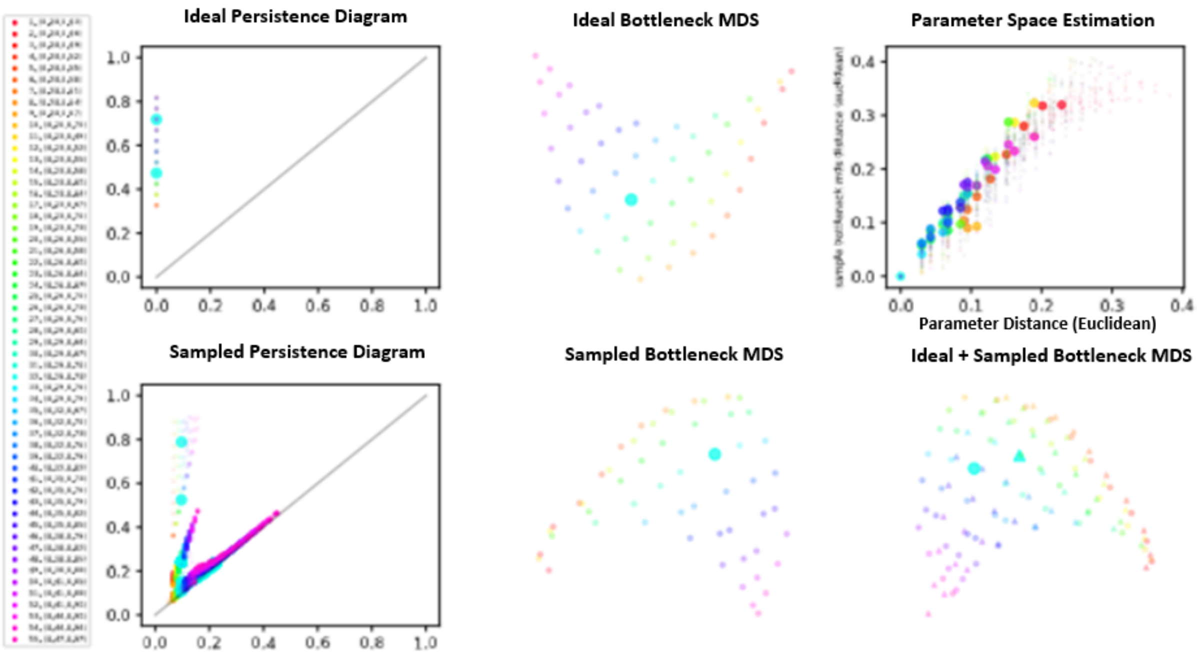


Figure 9: Plot of the pair $p_{32} = (0.29, 0.73)$ (aspect ratio ~ 2.52).

Figure 20, MDS of Pesistence Diagrams (p_3,2)

4.6 Novel Homological Methods for Sensor Network Integrity

The results of this portion of the effort are summarized in the report contained in Appendix IV.

4.7 Additional Approaches

This subsection presents the results of prototyping and simulating the less topological methods that were presented in Section 3.4.2 through Section 3.4.7.

4.7.1 Informativeness

This section tests the ability of the informativeness measure described in Section 3.4.2 to distinguish structured data sets from non-structured data sets. Two distance measures are utilized: Barelett's method and the Bures distance. First, Fig. 21 shows the result of Bartlett's method as applied to finding the distance of structured and non-strucutred matrices to the nearest non-informative matrix.

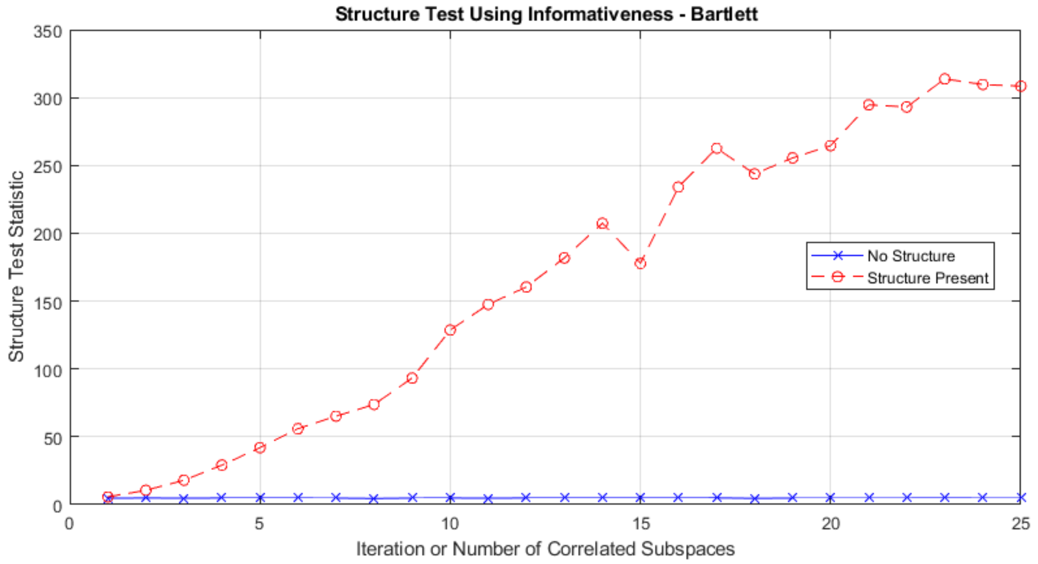


Figure 21: Informativeness with Bartlett's Method

Next, in Fig. 22 the Bures distance is applied to finding the distance of structured and non-strucutred matrices to the nearest non-informative matrix. Both the Barlett method and the Bures distance curves appear to deflect away from the non-structured cases as more correlated subspaces are added to the test data sets.

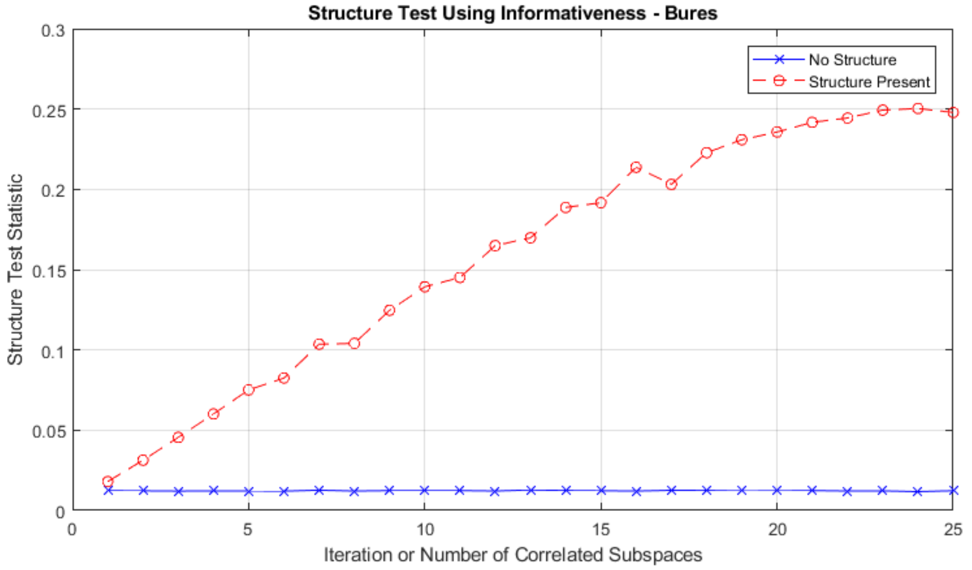
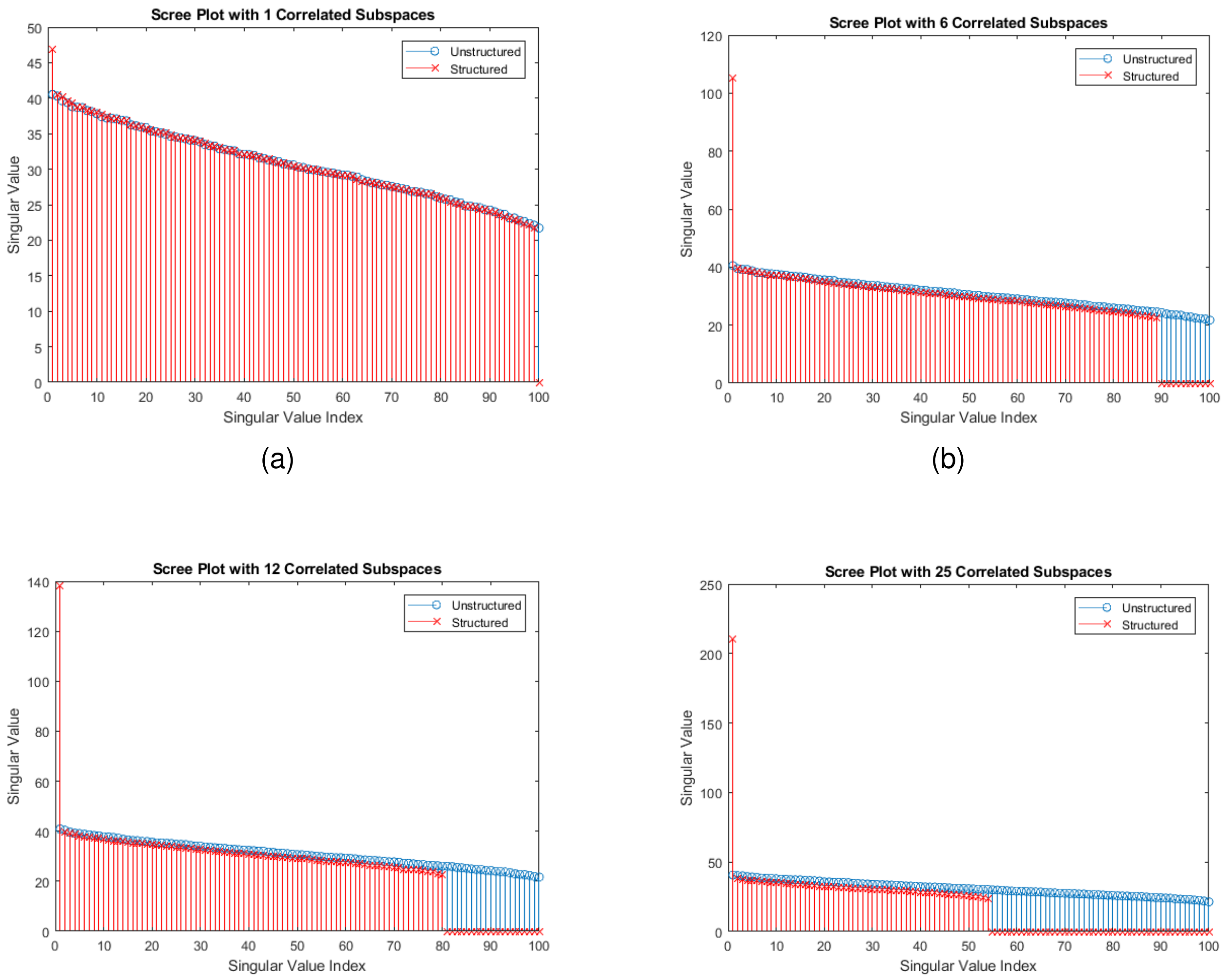


Figure 22: Informativeness with Bures Method

4.7.2 Scree-Plot Analysis

The scree plot discussed in Section 3.4.3 involves plotting the sorted singular values and analyzing the result profile. In this experiment, a scree profile is computed for both a structured and unstructured data set. The structure in the data set is represented by the number of significantly correlated dimensions or subspaces. The number of correlated subspaces is varied and the resulting profile is plotted. Values for the number correlated subspaces include $1, 6, 12$, and 25.

In Fig. 23a, a single pair of data vectors is correlated and the resulting singular value spectrum profile is plotted. Moving to Fig. 23b, the number of correlated components increases to 6. The effect on the scree plot is that the largest values is increased significantly, and a hard drop-off to zero is noticed at around index 90. Fig. Fig. 23c shows that as the number of correlated subspaces increases to 12, the largest singular value continues to increases, and the drop-off to zero decreases to about 80. Finally, in Fig. 23d, 25 dimensions of the data are now correlated and this has the effect of driving the largest singular value up further and continuing to decrease the drop-off to zero point to approximately 55. Overall, it appears that the more correlated components increases the largest singular value and decreases the index at which the singular values drop to zero.



(a) (b)

(c) (d)

Figure 23: Scree Plots for Various Levels of Structure.

4.7.3 Random Hadamard Multiplication

This section experiments with the method of point-wise multiplication of the data with a randomly generated matrix. The method is reviewed in Section 3.4.4, and is applied to structured and unstructured data. Fig. 24 depicts the result of iteratively computing Eq. 9 for a data set that is randomly generated with no structure, and also for a data set that contains an increasing number of significantly correlated components or subspaces. The number of subspaces was increased linearly from 1 to 25 over each iteration of the algorithm, and Eq. 9 was computed for the data set at each of the 25 iterations. The result indicates that the random Hadamard structure test statistic decreases monotonically with an increasing number of correlated subspaces. One benefit of this method is that no computationally expensive singular or eigenvalue decomposition is required.

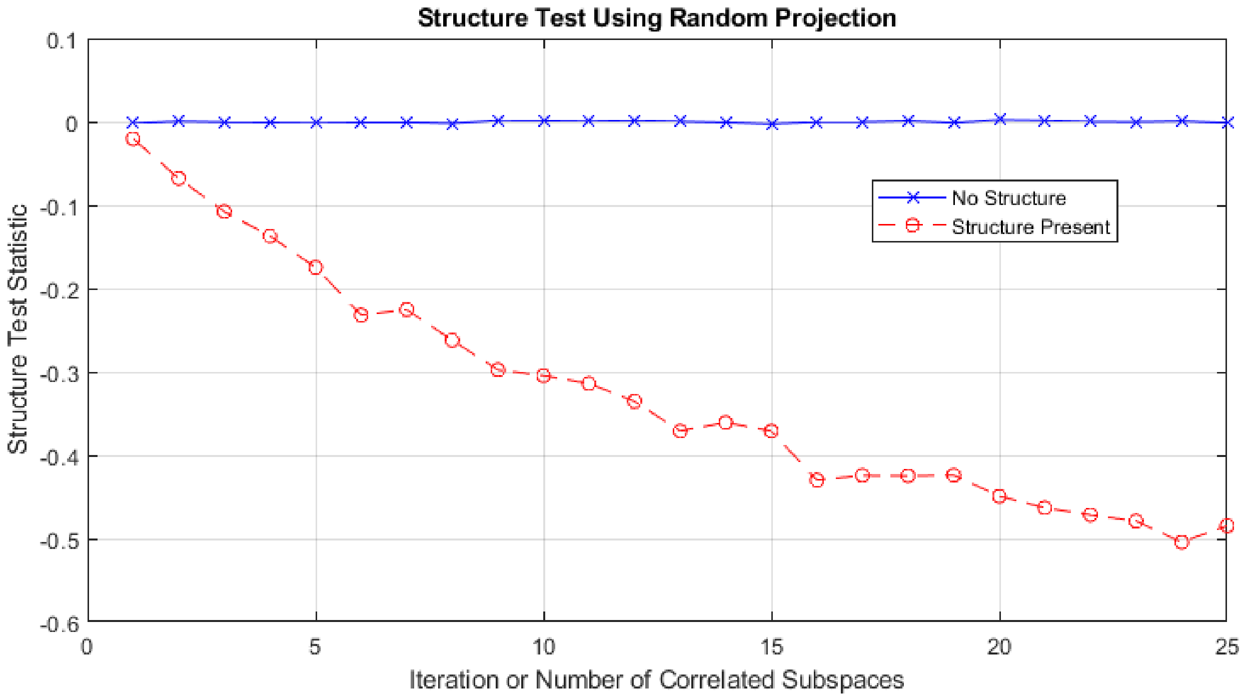


Figure 24: Random Hadamard Multiplication Structure Test Results

An additional experiment was performed to relate the random Hadamard structure test statistic to the off-diagonal elements of a normalized SPD matrix, i.e. a correlation matrix. The SPD matrix has constant off-diagonals; this constant is iterated from 0.025 to 0.5. This represents a series of non-informative matrices per Section 3.4.2. Therefore, since each SPD matrix is a non-informative matrix, each is at distance zero from the nearest non-informative matrix. At each iteration, as the off-diagonal constant element value is increased, the random Hadamard structure test statistic is computed from Eq. 9. The results are shown in Fig. 25.

Since each data matrix at each iteration is non-informative matrix, this experiment provides some idea as to how the random Hadamard strucutre test statistic innately reflects the magnitude of the correlations components, without regard (in some sense) for the information content of the data.

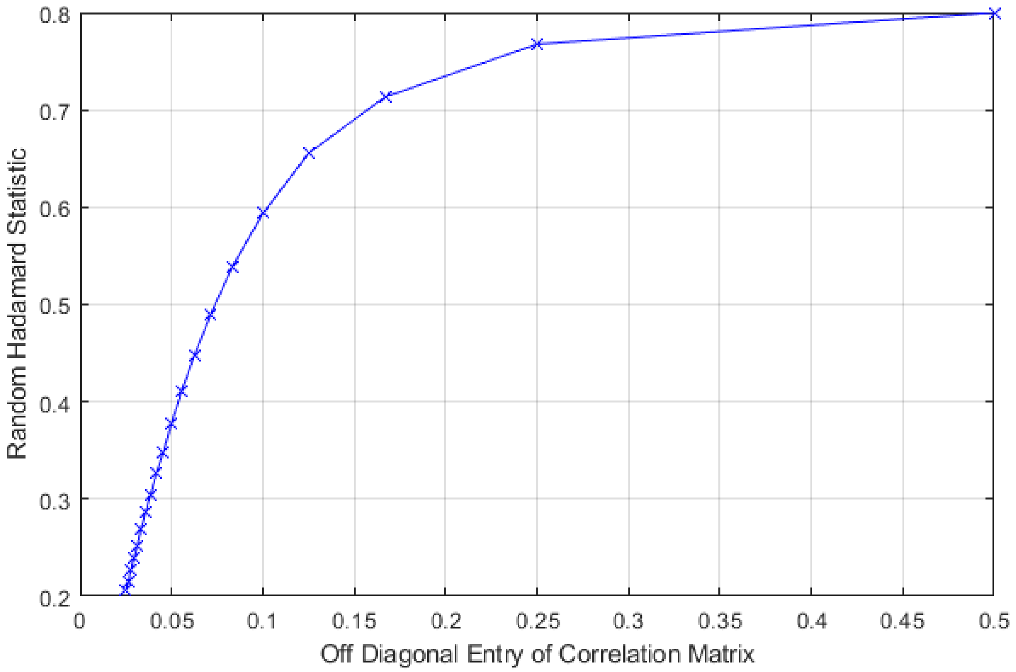


Figure 25: Off-Diagonal Element vs. Random Hadamard Structure Test

4.7.4 Shannon Entropy of Singular Values

Similar to the random Hadamard experiment, this section experiments with another method of structure detection, specifically computing the Shannon entropy of the set of singular values of the data matrix. The method is reviewed in Section 3.4.5, and is applied to structured and unstructured data. Fig. 26 depicts the result of iteratively computing Eq. 11 for a data set that is randomly generated with no structure, and also for a data set that contains an increasing number of significantly correlated components or subspaces. The number of subspaces was increased linearly from 1 to 25 over each iteration of the algorithm, and Eq. 11 was computed for the data set at each of the 25 iterations. The result indicates that the Shannon entropy of the set of singular values of the data can function as an effective structure test statistic. Like the random Hadamard statistic, the Shannon entropy decreases monotonically with an increasing number of correlated sub-spaces. However, unlike the random Hadamard statistic, the Shannon entropy of the singular values obviously requires a singular value decomposition, which can drive up the computational requirements.

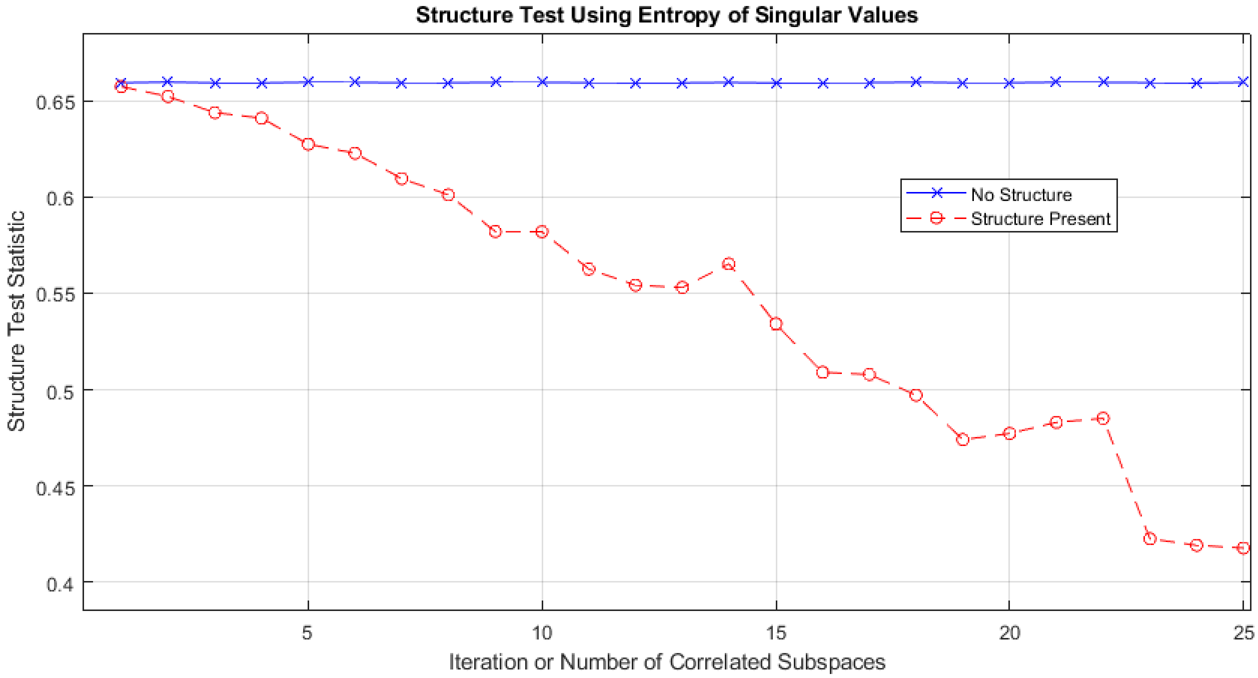


Figure 26: Shannon Entropy Structure Test Results

4.7.5 SPD Matrix Differences

In Section 3.4.7, the possibility of ascertaining the structure in a data set using covariance matrix distances between the covariance of the data set and the covariance of an identify matrix was discussed. This section looks at studying

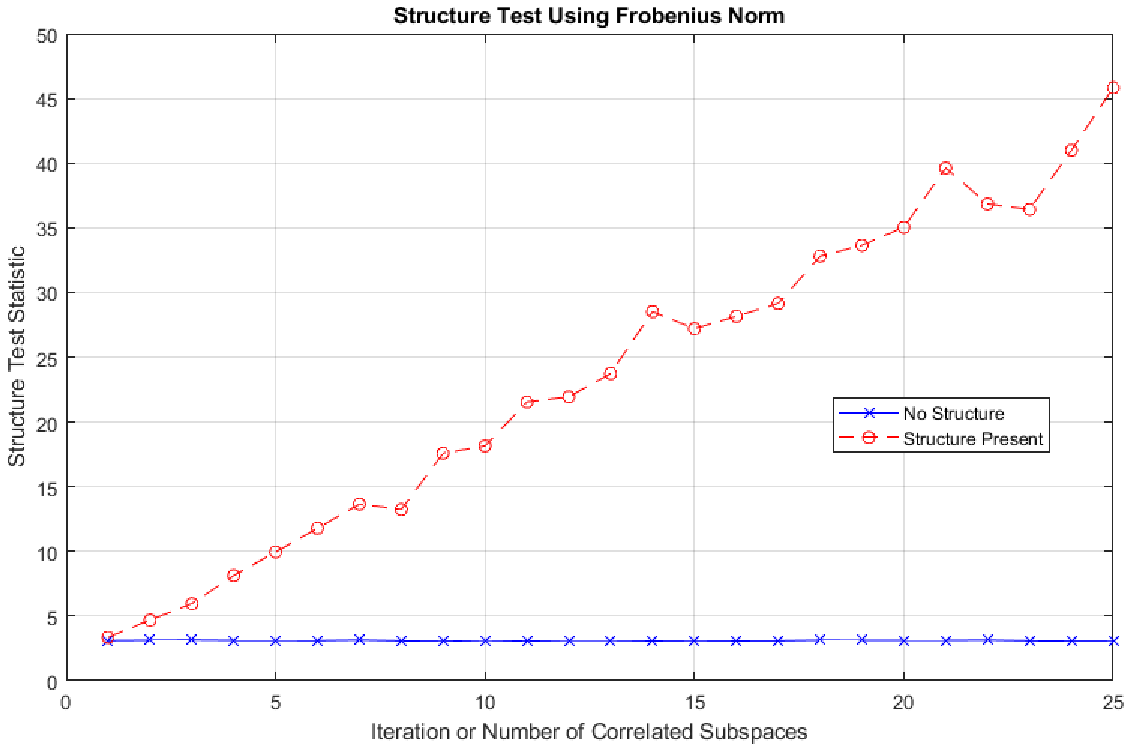


Figure 27: Frobenius Norm Structure Test Results

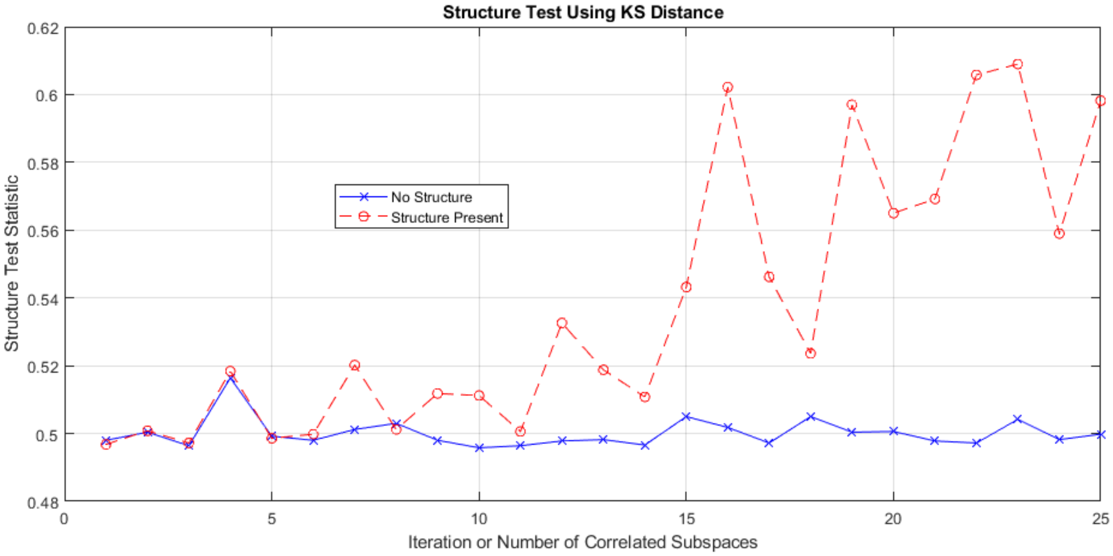


Figure 28: Kolmogorov-Smirnov Structure Test Results

As was discussed in Section 3.4.8, the length of the record length (the number of records or measurements or samples used to compute the covariance) can have a profound impact on the distance between two covariance matrices. To replicate the results of [15], Eq. (20) is plotted below, along with a curve derived from a Monte Carlo simulation that confirms the theoretical equation. Each covariance matrix is randomly constructed from a set of Gaussian distributed vectors. Each point on the graph represents the difference in Frobenius norm distance that can be expected due solely to record length, since each of the two data sets is drawn from the same Gaussian distribution. As can be seen, the optimimum (lowest) difference is in the middle of the curve, when the two data sets are of the same length.

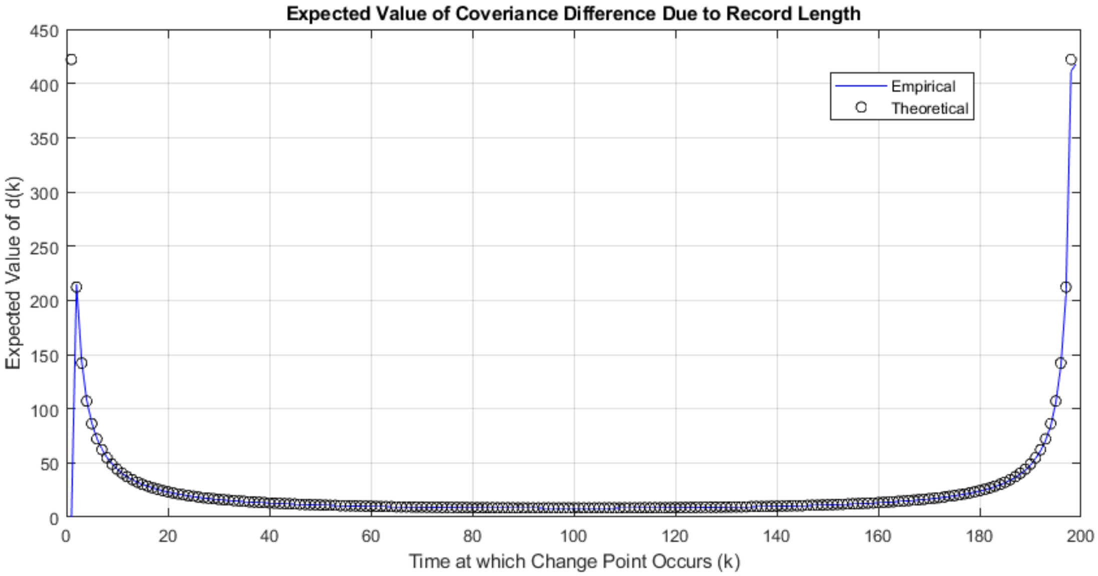


Figure 29: Record Length Impact on Frobenius Norm Covariance Distance

5 Conclusion

5.1 Summary

This report summarizes the results of the TOPEX project. The Topological Exploitation project (TOPEX) initially aimed at understanding and studying recent methods in topolog-ical data analysis (TDA). The research contained is intended to provide a basic under-standing of TDA and the type of signal information that TDA tools provide, with the aim of providing methods to facilitate unsupervised detection of structure within data. Addi-tionally, additional tools are investigated to perform structure detection in signal process-ing and data analysis (e.g. covariance distance measures and singular value analysis). Lastly, some high-level applications are explored. For example, the utility of TDA for op-portunistic data mining is briefly analyzed, along with some of the general issues relating to data mining.

5.2 Areas for Future Research

Euclidean distance matrices (EDM)and their application are an active field of research [25], [26]. EDMs are also PSD matrices. It would be useful to measure distances between EDMs to track them for changes. A recent publications do not appear to adequately address or view EDMs as points on a convex cone or Riemannian space. Regarding a sensor network application, it would also be useful to relate distances between two EDMs, each of which represents the distances among sensors at a given time, and relate that distance to the distance between two sets of sensor network measurements. This relationship could be used to track the changes in sensor network measurements over time.

Additionally, exploring the relationship between persistent homology and the concept of informativeness [2] would likely be useful; specifically how the two tools can be used to derive topological features from more traditional geometric tools. Lastly, a potential future endeavor may involve exploring the application of Hodge theory to statistically rank features of a data set [27].

5.3 Potential Commercial Applications

Some potential commercial applications of this work may include DSA spectrum report verification and falsification detection (see [21] and references contained therein), as well as biomedical signal analysis where sets of data need to be compared over vari-ous sessions, e.g., EEG data analysis [2], [28], [9]. It is hypothesized that the methods developed in this work may be applicable to mobile RF network analysis. For exam-ple, community detection within mobile networks has been studied by [29], where clique analysis is performed between two mobile networks and demonstrates that the density of the users affects the calling behavior significantly. Also, [30] explores a method for anomaly detection in mobile communication time series graphs. The method is based on an eigen-decomposition of the correlation matrix formed from the correlation network. In general, community detection within correlation has also been studied in various other contexts [31] [32], [33], [34], [35].

6 References

[1] H. Cheng, D. Feng, X. Shi, and C. Chen, “Data quality analysis and cleaning strategy for wireless sensor networks,” *EURASIP Journal on Wireless Communications and Networking*, vol. 61, 2018.

[2] A. Brockmeier, T. Mu, S. Ananiadou, and J. Goulermas, “Quantifying the informativeness of similarity measurements,” *Journal of Machine Learning Research*, 2017.

[3] D. Skillicorn, *Understanding Complex Data Sets: Data Mining with Matrix Decompositions*. Chapman and Hall, CRC Press, Taylor and Francis Group, 2007.

[4] S. Leatherdale, “Natural experiment methodology for research: a review of how different methods can support realworld research,” *International Journal of Social Research Methodology*, vol. 22, no. 1, pp. 19–35, 2018.

[5] Y. Yankelevsky and M. Elad, “Dictionary learning for high dimensional graph signals,” *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018.

[6] A. Sandryhaila and J. M. F. Moura, “Discrete signal processing on graphs: Graph fourier transform,” *ICASSP*, p. 6167–6170, 2013.

[7] D. Shuman, S. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, “The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains,” *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 83–98, 2013.

[8] C. Giusti and D. B. R. Ghrist, “Two’s company, three (or more) is a simplex: Algebraic-topological tools for understanding higher-order structure in neural data,” *Journal of Computational Neuroscience*, vol. 41, no. 1, pp. 1–14, 2016.

[9] H. Q. Minh and V. Murino, “Covariances in computer vision and machine learning,” in *Synthesis Lectures on Computer Vision* (G. Medioni and S. Dickinson, eds.), Morgan and Claypool Publishers, 2018.

[10] D. Richeson, *Euler’s Gem*. Princeton University Press, 2008.

[11] A. Zomorodian, *Topology for Computing*. Cambridge University Press, 2005.

[12] C. Giusti, E. Pastalkova, and C. C. V. Itskov, “Clique topology reveals intrinsic geometric structure in neural correlations,” *Proceedings of the National Academy of the Sciences of the United States of America*, vol. 112, no. 44, p. 455, 2015.

[13] M. A. Davenport, P. T. Boufounos, M. B. Wakin, and R. G. Baraniuk, “Signal processing with compressive measurements,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, pp. 445–460, April 2010.

[14] D. Barber, “Identifying graph clusters using variational inference and links to covariance parameterization,” *Philosophical Transactions of the Royal Society A*, 2017.

[15] I. Barnett and J. Onnela, “Change point detection in correlation networks,” *Scientific Reports*, vol. 6, no. 18893, 2016.

[16] G. Golub and C. V. Loan, eds., *Matrix Computations, Third Ed.* Johns Hopkins University Press, 1996.

[17] T. Anderson, “Asymptotic theory for principal component analysis,” *The Annals of Mathematical Statistics*, vol. 34, no. 1, pp. 122–148, 1963.

[18] D. Lawley, “On testing a set of correlation coefficients for equality,” *The Annals of Mathematical Statistics*, vol. 34, no. 1, pp. 149–151, 1963.

[19] M. Bartlett, “A note on the multiplying factors for various chi-square approximations,” *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 16, no. 2, p. 296–298, 1954.

[20] D. Achlioptas and F. McSherry, “Fast computation of low-rank matrix approximations,” *Journal of the ACM*, no. 2, 2007.

[21] J. Kelly, *Compressive Sensing Matrix Design and Evalutation with Cognitive Radio Applications*. PhD dissertation, State University of New York at Binghamton, 2019.

[22] P. Kvam and B. Vidakovic, eds., *Nonparametric Statistics with Applications to Science and Engineering*. John Wiley and Sons, Inc., 2007.

[23] C. P. A. Basu, H. Shioya, *Statistical Inference: The Minimum Distance Approach*. CRC Press, Taylor and Francis Group, 2011.

[24] R. Bhatia and J. Holbrook, “Riemannian geometry and matrix geometric means,” *Linear Algebra and its Applications*, vol. 413, pp. 594–618, 2006.

[25] J. Dattoro, *Convex Optimization Euclidean Distance Geometry Second Ed*. Meboo Publishing USA, 2017 http://meboo.convexoptimization.com/access.html (accessed 19-MAR2018).

[26] I. Dokmanic, R. Parhizkar, J. Ranieri, and M. Vertterli, “Euclidean distance matrices: Essential theory, algorithms, and applications,” *IEEE Signal Processing Magazine*, vol. 32, no. 6, pp. 12–30, 2015.

[27] X. Jiang, L. Lim, Y. Yao, and Y. Ye, “Statistical ranking and combinatorial hodge theory,” *Mathematical Programming, Series B*, pp. 203–244, 2011.

[28] O. Yair, M. Ben-Chen, and R. Talmon, “Parallel transport on the cone manifold of spd matrices for domain adaptation,” *IEEE Transactions on Signal Processing*, vol. 67, no. 7, pp. 1797–1811, 2019.

[29] M. Li and et al, “Communication cliques in mobile phone calling networks,” *Journal of Statistical Mechanics*, vol. 1, no. P11007, 2016.

[30] L. Akoglu and C. Faloutsos, “Event detection in time series of mobile communication graphs,” *Proceedings of the Army Science Conference*, pp. 77–79, 2010.

[31] J. O. et al, “Structure and tie strengths in mobile communication networks,” *Proceedings of the National Academy of Sciences USA*, vol. 104, no. 18, 2007.

[32] M. Hastings, “Community detection as an inference problem,” *Physical Review E*, 2006.

[33] L. Tang and H. Liu, *Community Detection and Mining in Social Media*. Morgan and Claypool Publishers, 2010.

[34] M. Newman, “Hypothesis testing for automated community detection in networks,” *Physical Review E*, 2006.

[35] P. Bickel and P. Sarkar, “Hypothesis testing for automated community detection in networks,” *Journal of the Royal Statistical Society Statistical Methodology Series B*, pp. 253–273, 2016.

[36] J. V. Michalowicz, J. M. Nichols, F. Bucholtz, and C. C. Olson, “An isserlis’ theorem for mixed gaussian variables: Application to the auto-bispectral density.,” *Journal of Statistical Physics*, vol. 136, no. 1, pp. 89 – 102, 2009.

[37] S. Kay, *Fundamentals of Statistical Signal Processing: Detection Theory*. Prentice Hall Publishers, 1998.

[38] B. Porat and B. Friedlander, “Computation of the exact information matrix of gaussian time series with stationary random components,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-34, no. 1, p. 118, 1986.

APPENDIX A: DERIVATION OF EXPECTED VALUE OF FROBENIUS NORM METRIC

This proof follows the outline that is presented in the supplemental material of [15]. Assume that the sampled data vector x_j is distributed according to a multi-variate normal distribution, and are stored in a matrix $X_j = [x_{1j}, ..., x_{nj}] \sim \mathcal{N}(0, \Sigma)$, with $j = 1, ..., L$, then the expected value of $d_{EUC}(k)$ for $k \in [2, ..., T-1]$ is given by

$$E\left[d_{EUC}(k)\right] = \left(\frac{1}{k} + \frac{1}{L-k}\right) \left(tr\left(\Sigma^2\right) + tr\left(\Sigma\right)^2\right), \tag{1}$$

where $d_{EUC}(k) = \|C(1,k) - C(k+1,L)\|_F^2$, and $C(i,j) = \frac{1}{j-i+1} \sum_{\ell=i}^{j} \left(X_\ell - \mu_X\right)\left(X_\ell - \mu_X\right)^T$.

The proof proceeds in two steps. First the distance corresponding to the Frobenius norm of the difference of two sample covariance matrices is found, and second the expected value of the distance is derived.

Frobenius Norm Metric of Sequential Covariance Matrices

The Frobenius distance of the difference between two covariance matrices can be written as

$$d_{EUC}(k) = \|C(1,k) - C(k+1,L)\|_F^2\,, \tag{2}$$

where $C(1,k)$ is the covariance matrix formed using data samples from 1 through k, i.e. $\{x_1, x_2, ..., x_k\}$, and $C(k+1,L)$ is the covariance matrix formed using samples $k+1$ through L, i.e. $\{x_{k+1}, x_{k+2}, ..., x_L\}$).

Since the Frobenius norm of a matrix M can be written as the trace of the matrix squared, i.e. $\|M\|_F = \left[tr\left\{M^T M\right\}\right]^{1/2}$, the Frobenius covariance distance can be rewritten using the trace operator as

$$\begin{aligned} d_{EUC}(k) &= \|C(1,k) - C(k+1,T)\|_F^2 \\ &= tr\left\{\left[C(1,k) - C(k+1,L)\right]^T\left[C(1,k) - C(k+1,L)\right]\right\} \\ &= tr\left\{\left[C(1,k) - C(k+1,L)\right]\left[C(1,k) - C(k+1,L)\right]\right\} \\ &= tr\left\{\left[C(1,k) - C(k+1,L)\right]^2\right\}. \end{aligned} \tag{3}$$

The last two lines of the preceding equation follow because C is a SPD matrix, and the difference of two symmetric matrices are again symmetric, therefore each factor in the preceding equation is symmetric. Expanding the square of the difference gives

$$\begin{aligned} d_{EUC}(k) &= tr\left\{\left[C(1,k) - C(k+1,L)\right]^2\right\} \\ &= tr\left\{C(1,k)C(1,k) - C(k+1,L)C(k+1,L) - 2C(1,k)C(k+1,L)\right\} \\ &= tr\left\{C(1,k)C(1,k)\right\} - tr\left\{C(k+1,T)C(k+1,L)\right\} - tr\left\{2C(1,k)C(k+1,L)\right\}. \end{aligned} \tag{4}$$

The last line follows from the linearity of the trace operator. At this point, recognize that the definition of C

$$\begin{aligned} C(i,j) &= \frac{1}{j-i-1}\sum_{\ell=i}^{j} X_\ell X_\ell^T \\ &= XD(i,j)X^T, \end{aligned} \tag{5}$$

can be substituted into each of the trace operations. The first trace operation is

$$\begin{aligned} tr\left\{C(1,k)C(1,k)\right\} &= tr\left\{\left(XD(1,k)X^T\right)\left(XD(1,k)X^T\right)\right\} \\ &= tr\left\{X^T X X^T X D(1,k)D(1,k)\right\} \\ &= tr\left\{\left(X^T X\right)^2 (D(1,k))^2\right\}, \end{aligned} \tag{6}$$

where the second and third lines follow from the invariance of the trace to permutations of symmetric matrices ($tr(ABC) = tr(ACB)$ for symmetric A, B, C). The second trace operation is

$$\begin{aligned} tr\left\{C(k+1,L)C(k+1,L)\right\} &= tr\left\{\left(XD(1,k+1)X^T\right)\left(XD(1,k+1)X^T\right)\right\} \\ &= tr\left\{X^T X X^T X D(1,k+1)D(1,k+1)\right\} \\ &= tr\left\{\left(X^T X\right)^2 (D(1,k+1))^2\right\}. \end{aligned} \tag{7}$$

Lastly, the third trace operation is evaluated as

$$\begin{aligned} tr\left\{2C(1,k)C(k+1,L)\right\} &= 2tr\left\{\left(XD(1,k)X^T\right)\left(XD(1,k+1)X^T\right)\right\} \\ &= 2tr\left\{X^T X X^T X D(1,k)D(1,k+1)\right\} \\ &= 2tr\left\{\left(X^T X\right)^2 0\right\} \\ &= 2 \cdot 0 = 0. \end{aligned} \tag{8}$$

Combining the results from evaluating each of the trace evaluations of Eqs. (6), (7), and (8) gives

$$\begin{aligned} d_{EUC}(k) &= tr\left\{C(1,k)C(1,k)\right\} - tr\left\{C(k+1,L)C(k+1,L)\right\} - tr\left\{2C(1,k)C(k+1,L)\right\} \\ &= tr\left\{\left(X^T X\right)^2 (D(1,k))^2\right\} - tr\left\{\left(X^T X\right)^2 (D(1,k+1))^2\right\} \\ &= tr\left\{\left(X^T X\right)^2 (D(1,k))^2 - \left(X^T X\right)^2 (D(1,k+1))^2\right\} \\ &= tr\left\{\left(X^T X\right)^2 \left((D(1,k))^2 - (D(1,k+1))^2\right)\right\}. \end{aligned} \tag{9}$$

Since the entries of the diagonal matrices are mutually exclusive, the last line can be written as

$$\begin{aligned} d_{EUC}(k) &= tr\left\{\left(X^TX\right)^2\left((D(1,k))^2-(D(1,k+1))^2\ \right)\right\} \\ &= tr\left\{\left(X^TX\right)^2(D(1,k)-D(1,k+1)\ )^2\right\} \\ &= tr\left\{\left(X^TX\left(D(1,k)-D(1,k+1)\right)\ \right)^2\right\} \\ &= tr\left\{\left(X^TX\Delta\ \right)^2\right\}. \end{aligned} \tag{10}$$

where Δ is defined as the element-wise difference between two covariance matrices at

$$\Delta \equiv D(1,k)-D(1,k+1). \tag{11}$$

Then the trace operation can be written using summation notation, resulting in

$$\begin{aligned} d_{EUC}(k) &= tr\left\{\left(X^TX\Delta\ \right)^2\right\} \\ &= \sum_{i=1}^{L}\sum_{j=1}^{L}\left(X_i^TX_j\right)^2\Delta_{jj}\Delta_{ii}. \end{aligned} \tag{12}$$

Now, separate the terms into those for which $i=j$ and those for which $i \neq j$ as

$$d_{EUC}(k) = \sum_{i=1}^{L}\left(X_i^TX_i\right)^2\Delta_{ii}^2 + \sum_{i=1}^{L}\sum_{j\in[1,L],j\neq i}\left(X_i^TX_j\right)^2\Delta_{jj}\Delta_{ii}. \tag{13}$$

Expected Value of Frobenius Norm Metric Now proceed to take the expected value of $d_{EUC}(k)$ as

$$
\begin{aligned}
E\left[d_{EUC}(k)\right] &= E\left[\sum_{i=1}^{T}\left(X_i^T X_i\right)^2 \Delta_{ii}^2 \; + \; \sum_{i=1}^{T} \sum_{j\in[1,T],j\neq i} \left(X_i^T X_j\right)^2 \Delta_{jj}\Delta_{ii}\right] \\
&= E\left[\sum_{i=1}^{T}\left(X_i^T X_i\right)^2 \Delta_{ii}^2\right] + E\left[\sum_{i=1}^{T} \sum_{j\in[1,T],j\neq i} \left(X_i^T X_j\right)^2 \Delta_{jj}\Delta_{ii}\right] \\
&= \sum_{i=1}^{T} E\left[\left(X_i^T X_i\right)^2\right] \Delta_{ii}^2 + \sum_{i=1}^{T} \sum_{j\in[1,T],j\neq i} E\left[\left(X_i^T X_j\right)^2\right] \Delta_{jj}\Delta_{ii} \\
&= \underbrace{\sum_{i=1}^{T} \Delta_{ii}^2 E\left[\left(X_i^T X_i\right)^2\right]}_{Term\,1} + \underbrace{\sum_{i=1}^{T} \sum_{j\in[1,T],j\neq i} \Delta_{jj}\Delta_{ii} E\left[\left(X_i^T X_j\right)^2\right]}_{Term\,2}
\end{aligned}
\tag{14}
$$

From this point, *Term 1* and *Term 2* are each further divided into terms prior to the index k and those terms at $k+1$ and later.

Term 1

Recall that $D(i,j)$ is defined to be the diagonal matrix

$$
D(i,j)_{\ell\ell} = \begin{cases} 1/(j-i+1) & i \leq \ell \leq j \\ 0 & otherwise. \end{cases}
\tag{15}
$$

Written out, the matrices from $i = 1$ to $j = k$ and from $i = k+1$ to $j = T$ are, respectively

$$
D(1,k) = \begin{bmatrix}
\frac{1}{k} & 0 & 0 & 0 & 0 & \cdots & 0 \\
0 & \frac{1}{k} & 0 & 0 & 0 & \cdots & 0 \\
0 & 0 & \ddots & 0 & 0 & \cdots & 0 \\
0 & 0 & 0 & \frac{1}{k} & 0 & \cdots & 0 \\
0 & 0 & 0 & 0 & 0 & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & 0 & 0 & \cdots & 0
\end{bmatrix}
\tag{16}
$$

$$
D(k+1,T) = \begin{bmatrix}
0 & 0 & \cdots & 0 & 0 & 0 & 0 \\
0 & 0 & \cdots & 0 & 0 & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & \cdots & 0 & 0 & 0 & 0 \\
0 & 0 & \cdots & 0 & \frac{1}{T-k} & 0 & 0 \\
0 & 0 & \cdots & 0 & 0 & \ddots & 0 \\
0 & 0 & \cdots & 0 & 0 & 0 & \frac{1}{T-k}
\end{bmatrix},
\tag{17}
$$

so that the first k diagonal elements of $D(1,k$ are equal to $\frac{1}{k}$ with the rest of the entries equal to zero, and the last $L-k$ diagonal entries of $D(k+1,L)$ are equal to $\frac{1}{L-k}$ with the rest of the entries equal to zero. E.g., for $i=1$, $j=k=4$ and $T=7$

$$D(1,4)=\begin{bmatrix}\frac{1}{4} & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & \frac{1}{4} & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & \frac{1}{4} & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & \frac{1}{4} & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0\end{bmatrix}, \tag{18}$$

and

$$D(5,7)=\begin{bmatrix}0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & \frac{1}{3} & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & \frac{1}{3} & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{3}\end{bmatrix}. \tag{19}$$

Therefore, $D(1,k)$ is equal to $\frac{1}{k-1+1}=\frac{1}{k}$ on the first k diagonal entries, and zero everywhere else. Correspondingly, $D(k+1,T)$ is equal to $\frac{1}{T-(k+1)+1}=\frac{1}{T-k}$ on the last $T-k$ diagonal entries and zero everywhere else. As a result,

$$\Delta_{ii}=D_{ii}(1,k)-D_{ii}(k+1,T)=\begin{cases}1/k & i\leq k\\ -1/(T-k) & k+1\leq i\leq T.\end{cases} \tag{20}$$

E.g., written out for $T=7$ and $k=4$, the matrix $\Delta=D(1,k)-D(k+1,T)$ looks like

$$\Delta=D(1,4)-D(5,7)=\begin{bmatrix}\frac{1}{4} & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & \frac{1}{4} & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & \frac{1}{4} & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & \frac{1}{4} & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & \frac{-1}{3} & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & \frac{-1}{3} & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{-1}{3}\end{bmatrix} \tag{21}$$

Therefore in general, the square Δ_{ii}^2 takes on only two values through the sampling interval $[1,T]$, as expressed by

$$\Delta_{ii}^2=\begin{cases}1/k^2 & i\leq k\\ 1/(T-k)^2 & k+1\leq i\leq T.\end{cases} \tag{22}$$

Therefore, prior to the change at sample k, Δ_{ii}^2 is constant and equals $1/k^2$. After the change (from $k+1$ to T), Δ_{ii}^2 is again constant, but equal to $1/(T-k)^2$. Since there are $'k'$ samples prior to the change, and $T-k$ samples after the change, the summation over all of *T* can be expressed as

$$\begin{aligned}\sum_{i=1}^{T} \Delta_{ii}^2 &= k \cdot 1/k^2 + (T-k) \cdot 1/(T-k)^2 \\ &= \frac{1}{k} + \frac{1}{T-k}.\end{aligned} \tag{23}$$

Therefore, substituting in for Δ_{ii}^2, *Term 1* in Eq. (14) can be written as

$$\begin{aligned}Term\ 1 &= \sum_{i=1}^{T} \Delta_{ii}^2 E\left[\left(X_i^T X_i\right)^2\right] \\ &= \left(\frac{1}{k} + \frac{1}{T-k}\right) E\left[\left(X_i^T X_i\right)^2\right].\end{aligned} \tag{24}$$

Now the expected value in the last line can be evaluated as $2tr(\Sigma^2) + tr(\Sigma)^2$, giving

$$Term\ 1 = \left(\frac{1}{k} + \frac{1}{T-k}\right) E\left[\left(X_i^T X_i\right)^2\right]. \tag{25}$$

Term 2

For *Term 2* of Eq. (14),

$$\sum_{i=1}^{T} \sum_{j \in [1,T], j \neq i} \Delta_{jj} \Delta_{ii} E\left[\left(X_i^T X_j\right)^2\right], \tag{26}$$

begin by separating Term 2 into two sums: one sum representing the first k terms *prior* the changepoint, and the second sum representing the contribution from the last $T-k$ terms *after* the changepoint.

The pre-change sum *Term 2* is written as

$$\sum_{i=1}^{k} \sum_{j \in [1,k], j \neq i} \Delta_{jj} \Delta_{ii} E\left[\left(X_i^T X_j\right)^2\right]. \tag{27}$$

Here, $\Delta_{ii} = D(1,k) - 0_{nxn}$ and $\Delta_{jj} = D(1,k-1) - D(k,k)$ so that

$$\sum_{i=1}^{k} \sum_{j\in[1,k],j\neq i} \Delta_{ii}\Delta_{jj} E\left[\left(X_i^T X_j\right)^2\right] = \sum_{i=1}^{k} \left((k-1)\frac{1}{k^2} - \frac{1}{k}\right) E\left[\left(X_i^T X_j\right)^2\right]. \tag{28}$$

Evaluating the summation over the index *k* results in the pre-change sum

$$\sum_{i=1}^{k} \left((k-1)\frac{1}{k^2} - \frac{1}{k}\right) E\left[\left(X_i^T X_j\right)^2\right] = k\left(\frac{k-1}{k^2} - \frac{1}{k}\right) E\left[\left(X_i^T X_j\right)^2\right]. \tag{29}$$

The post-change sum for *Term 2* is written as

$$\sum_{i=k+1}^{T} \sum_{j\in[k+1,T],j\neq i} \Delta_{jj}\Delta_{ii} E\left[\left(X_i^T X_j\right)^2\right]. \tag{30}$$

Here, $\Delta_{ii} = D(1,k) - 0_{nxn}$ and $\Delta_{jj} = D(1,k-1) - D(k,k)$ so that

$$\sum_{i=k+1}^{T} \sum_{j\in[k+1,T],j\neq i} \Delta_{jj}\Delta_{ii} E\left[\left(X_i^T X_j\right)^2\right] = \sum_{i=k+1}^{T} \left(\frac{T-k-1}{(T-k)^2} - \frac{1}{T-k}\right) E\left[\left(X_i^T X_j\right)^2\right]. \tag{31}$$

Evaluating the summation over the index *k* results in the post-change sum

$$\sum_{i=k+1}^{T} \left(\frac{T-k-1}{(T-k)^2} - \frac{1}{T-k}\right) E\left[\left(X_i^T X_j\right)^2\right] = (T-k)\left(\frac{T-k-1}{(T-k)^2} - \frac{1}{T-k}\right) E\left[\left(X_i^T X_j\right)^2\right]. \tag{32}$$

Combining the pre- and post-change parts of $Term2$ gives

$$\begin{aligned} Term\,2 &= k\left(\frac{k-1}{k^2} - \frac{1}{k}\right) E\left[\left(X_i^T X_j\right)^2\right] + (T-k)\left(\frac{T-k-1}{(T-k)^2} - \frac{1}{T-k}\right) E\left[\left(X_i^T X_j\right)^2\right] \\ &= \left(k\left(\frac{k-1}{k^2} - \frac{1}{k}\right) + (T-k)\left(\frac{T-k-1}{(T-k)^2} - \frac{1}{T-k}\right)\right) E\left[\left(X_i^T X_j\right)^2\right] \\ &= \left(\frac{k-1}{k} - 1 + \frac{T-k-1}{(T-k)} - 1\right) E\left[\left(X_i^T X_j\right)^2\right] \\ &= \left(\frac{k-1}{k} + \frac{T-k-1}{(T-k)} - 2\right) E\left[\left(X_i^T X_j\right)^2\right] \\ &= -\left(\frac{1}{k} + \frac{1}{(T-k)}\right) E\left[\left(X_i^T X_j\right)^2\right] \end{aligned} \tag{33}$$

At this point, the expected value in the last line can be evaluated as $tr(\Sigma^2)$, giving

$$Term\,2 = -\left(\frac{1}{k} + \frac{1}{(T-k)}\right) E\left[\left(X_i^T X_j\right)^2\right] \tag{34}$$

Combine Term 1 and Term 2

Substituting in Term 1 (Eq.(25)) and Term 2 (Eq.(34)) into Equation (14) gives

$$\begin{aligned}
E\left[d_{EUC}(k)\right] &= \underbrace{\sum_{i=1}^{T} \Delta_{ii}^2 E\left[\left(X_i^T X_i\right)^2\right]}_{Term\,1} + \underbrace{\sum_{i=1}^{T} \sum_{j\in[1,T],j\neq i} \Delta_{jj}\Delta_{ii} E\left[\left(X_i^T X_j\right)^2\right]}_{Term\,2} \\
&= \underbrace{\left(\frac{1}{k} + \frac{1}{T-k}\right)\left(E\left[\left(X_i^T X_i\right)^2\right]\right)}_{Term\,1} - \underbrace{\left(\frac{1}{k} + \frac{1}{T-k}\right)\left(E\left[\left(X_i^T X_j\right)^2\right]\right)}_{Term\,2} \\
&= \left(\frac{1}{k} + \frac{1}{T-k}\right)\left(E\left[\left(X_i^T X_i\right)^2\right] - E\left[\left(X_i^T X_j\right)^2\right]\right)
\end{aligned} \tag{35}$$

for $i \neq j$, where the last line follows from factoring out the $\frac{1}{k} + \frac{1}{T-k}$ from each term. Now, it remains to evaluate the two expected value terms. The two expected value terms each represent fourth-order moments of random vectors that are drawn from a zero-mean multivariate Gaussian distribution. The proof will proceed by evaluating the case for $i \neq j$, then move on to address the $i = j$ case.

Covariances $E\left[\left(X_i^T X_j\right)^2\right]$

Writing out the expected value for the cross-term first gives

$$\begin{aligned}
E\left[\left(X_i^T X_j\right)^2\right] &= E\left[X_i^T X_j X_i^T X_j\right] \\
&= E\left[\sum_{m,n,q,r} X_i(m) I_{mn} X_j(n) X_i(q) I_{pq} X_j(r)\right] \\
&= \sum_{m,n,q,r} I_{mn} I_{pq} E\left[X_i(m) X_j(n) X_i(q) X_j(r)\right].
\end{aligned} \tag{36}$$

The last line follows from the linearity of the expectation operation. To evaluate this, Isserlis' theorem can be applied [36]. In general, it states that the expected value of a product of $2n$ zero-mean Gaussian multivariate random vectors y_i can be found as [36]

$$E\left[y_1 y_2 y_3 \cdots y_{2n}\right] = \sum \prod E\left[y_i y_j\right] \\ = \sum \prod Cov(y_i y_j). \tag{37}$$

In particular, for the expectation $E\left[X_i^T X_j X_i^T X_j\right]$, Isserlis' theorem gives [36–38]

$$\begin{aligned} E\left[X_i(m)X_j(n)X_i(q)X_j(r)\right] = & E\left[X_i(m)X_j(n)\right]E\left[X_i(q)X_j(r)\right]+ \\ & E\left[X_i(m)X_i(q)\right]E\left[X_j(n)X_j(r)\right]+ \\ & E\left[X_i(m)X_j(r)\right]E\left[X_j(n)X_i(q)\right] \end{aligned} \tag{38}$$

$$= 0 + \sigma_{mq}\sigma_{nr} + 0,$$

Substituting this into the formula above gives

$$\begin{aligned} E\left[\left(X_i^T X_j\right)^2\right] &= \sum_{m,n,q,r} I_{mn} I_{pq} E\left[X_i(m)X_j(n)X_i(q)X_j(r)\right] \\ &= \sum_{m,n,q,r} I_{mn} I_{pq} \sigma_{mq}\sigma_{nr} \end{aligned} \tag{39}$$

Since I_{mn} is non-zero only when $m = n$ and I_{qr} is non-zero only when $q = r$ the above can be rewritten as

$$\begin{aligned} E\left[\left(X_i^T X_j\right)^2\right] &= \sum_{m,n,q,r} I_{mn} I_{pq} \sigma_{mq}\sigma_{nr} \\ &= \sum_{m,n} \sigma_{mn}\sigma_{mn} \\ &= tr\left(\Sigma^2\right) \end{aligned} \tag{40}$$

Since for this case $i \neq j$, and X_i is drawn independently from X_j, the cross-terms $\sigma_{ij} = 0$, giving

$$\begin{aligned} E\left[X_i^T X_j X_i^T X_j\right] &= 2\sum_m \sigma_{ij}^2 + \sum_m \sigma_{ii}\sigma_{jj} \\ &= \sum_m \sigma_{ii}\sigma_{jj}. \end{aligned} \tag{41}$$

Variances $E\left[\left(X_i^T X_i\right)^2\right]$

$$E\left[\left(X_i^T X_i\right)^2\right] = \sum_{m,n,q,r} I_{mn} I_{pq} E\left[X_i(m)X_i(n)X_i(q)X_i(r)\right]. \tag{42}$$

In particular, for the expectation $E\left[X_i^T X_i X_i^T X_i\right]$, Isserlis' theorem gives

$$\begin{aligned} E\left[X_i(m)X_i(n)X_i(q)X_i(r)\right] = {} & E\left[X_i(m)X_i(n)\right]E\left[X_i(q)X_i(r)\right] + \\ & E\left[X_i(m)X_i(q)\right]E\left[X_i(n)X_i(r)\right] + \\ & E\left[X_i(m)X_i(r)\right]E\left[X_i(n)X_i(q)\right] \end{aligned} \tag{43}$$

$$= \sigma_{mn}\sigma_{qr} + \sigma_{mq}\sigma_{nr} + \sigma_{mr}\sigma_{nq},$$

Substituting this into the formula above gives

$$\begin{aligned} E\left[\left(X_i^T X_i\right)^2\right] &= \sum_{m,n,q,r} I_{mn} I_{qr} E\left[X_i(m)X_i(n)X_i(q)X_i(r)\right] \\ &= \sum_{m,n,q,r} I_{mn} I_{qr}\left(\sigma_{mn}\sigma_{qr} + \sigma_{mq}\sigma_{nr} + \sigma_{mr}\sigma_{nq}\right) \\ &= \sum_{m,n,q,r} I_{mn} I_{qr}\left(\sigma_{mn}\sigma_{qr}\right) + \sum_{m,n,q,r} I_{mn} I_{qr}\left(\sigma_{mq}\sigma_{nr}\right) + \sum_{m,n,q,r} I_{mn} I_{qr}\left(\sigma_{mr}\sigma_{nq}\right) \\ &= \sum_{m,n} I_{mn}\sigma_{mn} \sum_{q,r} I_{qr}\sigma_{qr} + \sum_{m,n,q,r} I_{mn} I_{qr}\left(\sigma_{mq}\sigma_{nr}\right) + \sum_{m,n,q,r} I_{mn} I_{qr}\left(\sigma_{mr}\sigma_{nq}\right) \\ &= tr\left(\Sigma\right) tr\left(\Sigma\right) + tr\left(\Sigma^2\right) + tr\left(\Sigma^2\right) \\ &= tr\left(\Sigma\right)^2 + 2tr\left(\Sigma^2\right) \end{aligned} \tag{44}$$

$$\begin{aligned} E\left[d_{EUC}(k)\right] &= \left(\frac{1}{k} + \frac{1}{T-k}\right)\left(E\left[\left(X_i^T X_i\right)^2\right]\right) - \left(\frac{1}{k} + \frac{1}{T-k}\right)E\left[\left(X_i^T X_j\right)^2\right] \\ &= \left(\frac{1}{k} + \frac{1}{T-k}\right)\left(2tr(\Sigma^2) + tr(\Sigma)^2\right) - \left(\frac{1}{k} + \frac{1}{T-k}\right)tr(\Sigma^2) \\ &= \left(\frac{1}{k} + \frac{1}{T-k}\right)\left(tr(\Sigma^2) + tr(\Sigma)^2\right). \end{aligned} \tag{45}$$

APPENDIX B: DISTRIBUTIONS OF OFF-DIAGONAL CORRELATION MATRIX Probability Mass and Distribution Functions First, two basic statistical functions are now defined. Let $X \equiv X_1, X_2, \ldots X_N$ be a set of samples from a discrete distribution. The first statistical function to be defined is the *probability mass function* (PMF) or empirical probability density

$$f(x_i) = \frac{1}{n} \sum_{i=1}^{n} \mathtt{I}(x_i \leq X_i < x_{i+1}), \tag{46}$$

where $[x_i, x_{i+1}]$ is the i^{th} sub-interval or bin, and there are n bins altogether.

The second tool to define is the *cumulative distribution function (CDF)* (or empirical distribution function (EDF) in the discrete case) of the entries of the Gram matrix of the effective dictionary. The CDF is defined as follows and is adapted from [22]. Again, let $X \equiv X_1, X_2, \ldots X_n$ be a set of samples from a discrete distribution. Then, the empirical distribution function (EDF) of the sample set is

$$F(x) = \frac{1}{n} \sum_{i=1}^{n} \mathtt{I}(X_i \leq x). \tag{47}$$

where $\mathtt{I}(\rho)$ is the *indicator function*, which is equal to 1 if the condition ρ is true, and equal to 0 if the condition ρ is false, X_i is the subset of n samples from X that are less than or equal to x.

APPENDIX C: TOPOLOGICAL BACKGROUND

A Brief History of Topology

As mathematicians researched surfaces in one, two, three, and higher dimensions, they looked for invariants among different classes of sur-faces, i.e. coarse measures that would distinguish one surface from another. One stellar example of this is Eulers polyhedron formula relating vertices, edges and sides of a poly-hedron. Further research extended Euler's formula to shapes other than polyhedra, e.g. to non-convex polyhedra, to graphs, etc. This signified an historical shift in thinking about the geometry of shapes to the topology of shapes. E.g., how far can a polyhedron be altered, bent, twisted, stretched such that it continues to obey Euler's formula?

In an attempt to address such questions, mathematicians began to study new aspects of surfaces such as *orientability*, as well as how to tell whether two surfaces were 'the same' as each other, i.e. classification of surfaces. I.e. whether they are homeomorphic. "'Two surfaces are homeomorphic if there exists a one-to-one correspondence"' between the points in the objects, such that the correspondence "'preserves *closeness* "' [10]. Points that are clustered in one shape are also clustered in the other shape [10].

Simplicial Complexes A simplicial complex is a means to represent relationships among sets of data points. The following definitions are adapted from [8]. A simplicial complex consists of a pair of sets:

- Vertices
- Simplices

Homology Homology is a principled numerical technique for computing the connectedness of an object. The homology of an object can be described qualitatively by describing it in terms of the dimension of the void that the cycle encloses.

At each threshold of the filtration, three homological quantities are computed:

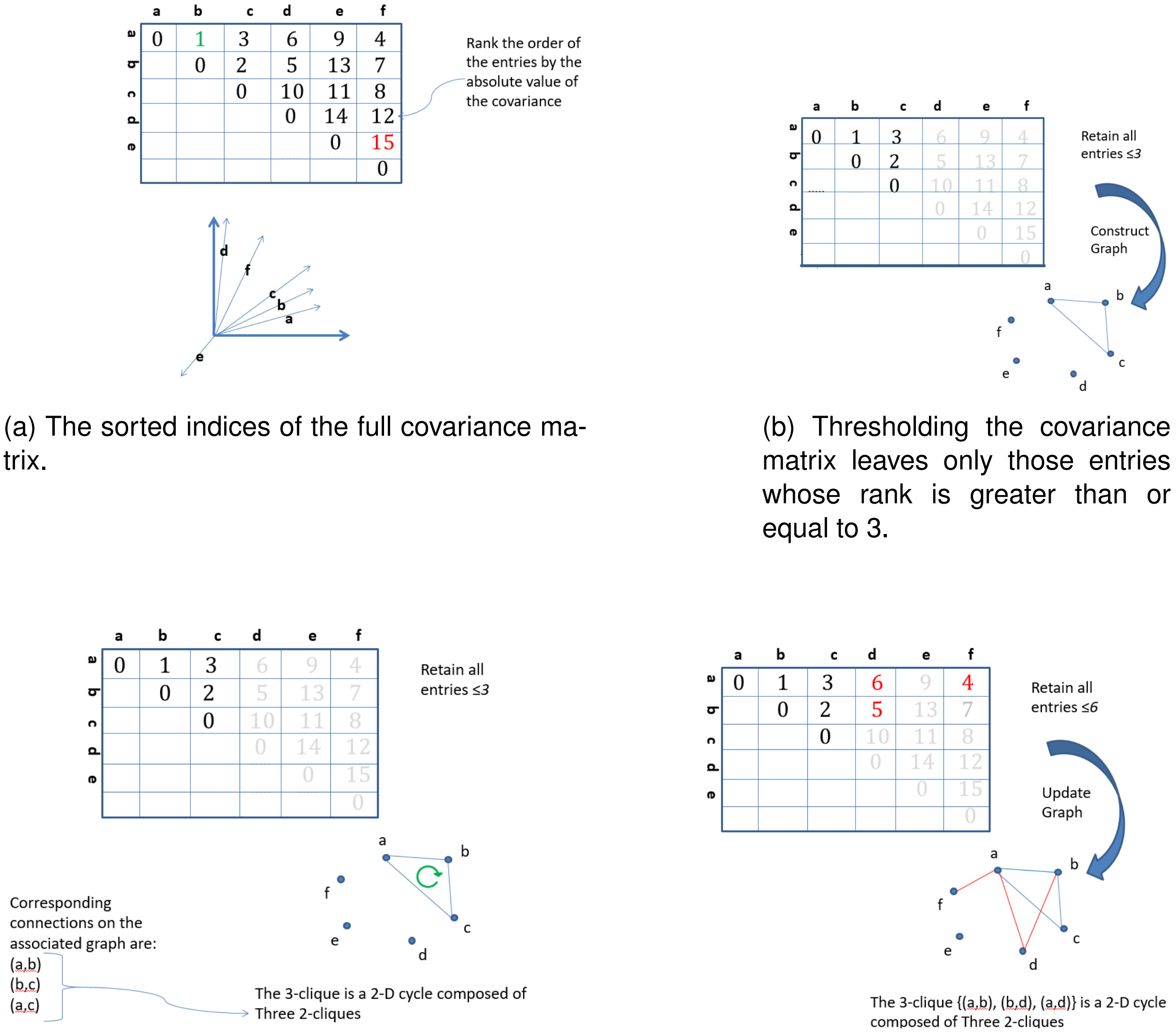
- Number of 0-Dimensional Cycles (Connected Components)
- Number of 1-Dimensional Cycles (Cycles)
- Number of 2-Dimensional Cycles (Cavities)

Persistent Homology

Fig. depicts a toy example of the basic steps that are involved in computing homology. First, begin with the upper triangular portion of a covariance matrix and sort the entries from lowest to highest absolute value. In Fig. 30a, a matrix is labeled with its entires rank, in order from lowest (1) in green to highest (15) in red. At this point, the actual entries of the covariance matrix are inconsequential, so long as the order of the entries is identified. The three-dimensional vectors below the matrix are one possible representation of the vectors from which the actual covariance matrix is derived.

Begin the thresholding process by retaining only those entries of the covariance matrix whose rank is greater than or equal to 3. The rank order of the corresponding thresholded covariance matrix entries that remain after thresholding is shown in black in Fig. 30b. The remainder of the entries are greyed out. Based on the location of the remaining entries in the matrix, begin to construct an undirected graph, where each of the nodes represents each column and row of the covariance matrix. On the graph, connect the nodes that correspond to the respective row and column of each entry that remains after thresholding. For the step in Fig. 30b, on of the remaining entries is entry 1 whose row and column index are (a, b). Therefore, connect node a to node b in the graph. Similarly, entry 2 remains and its location in the matrix is given by (b, c), so connect node b to node c in the graph. Lastly, entry three also remains after thresholding, and since its location in the matrix is (a, c), connect node a to node c in the graph.

At this threshold, the graph forms one loop or 2-dimensional cycle from nodes a, b, c as shown in Fig. 30c, and in particular it forms a 3-clique since every node in a, b, c is connected to every other node in a, b, c. Next, a higher threshold is selected, e.g. retain only those entries in the orginal covariance matrix that are less than or equal to 6, as shown in Fig. 30d. Based on the location of the new set of thresholded entries that are retained, the graph is updated and now contains an additional 3-clique from the connections among nodes a, b, d and a new 2-clique from the conection between nodes a and f.



	a	b	c	d	e	f
a	0	1	3	6	9	4
b		0	2	5	13	7
c			0	10	11	8
d				0	14	12
e					0	15
						0

Rank the order of the entries by the absolute value of the covariance

(a) The sorted indices of the full covariance matrix.

	a	b	c	d	e	f
a	0	1	3	6	9	4
b		0	2	5	13	7
c			0	10	11	8
d				0	14	12
e					0	15
						0

Retain all entries ≤3

Construct Graph

(b) Thresholding the covariance matrix leaves only those entries whose rank is greater than or equal to 3.

	a	b	c	d	e	f
a	0	1	3	6	9	4
b		0	2	5	13	7
c			0	10	11	8
d				0	14	12
e					0	15
						0

Retain all entries ≤3

Corresponding connections on the associated graph are:
(a,b)
(b,c)
(a,c)

The 3-clique is a 2-D cycle composed of Three 2-cliques

(c) Compute Cycles

	a	b	c	d	e	f
a	0	1	3	6	9	4
b		0	2	5	13	7
c			0	10	11	8
d				0	14	12
e					0	15
						0

Retain all entries ≤6

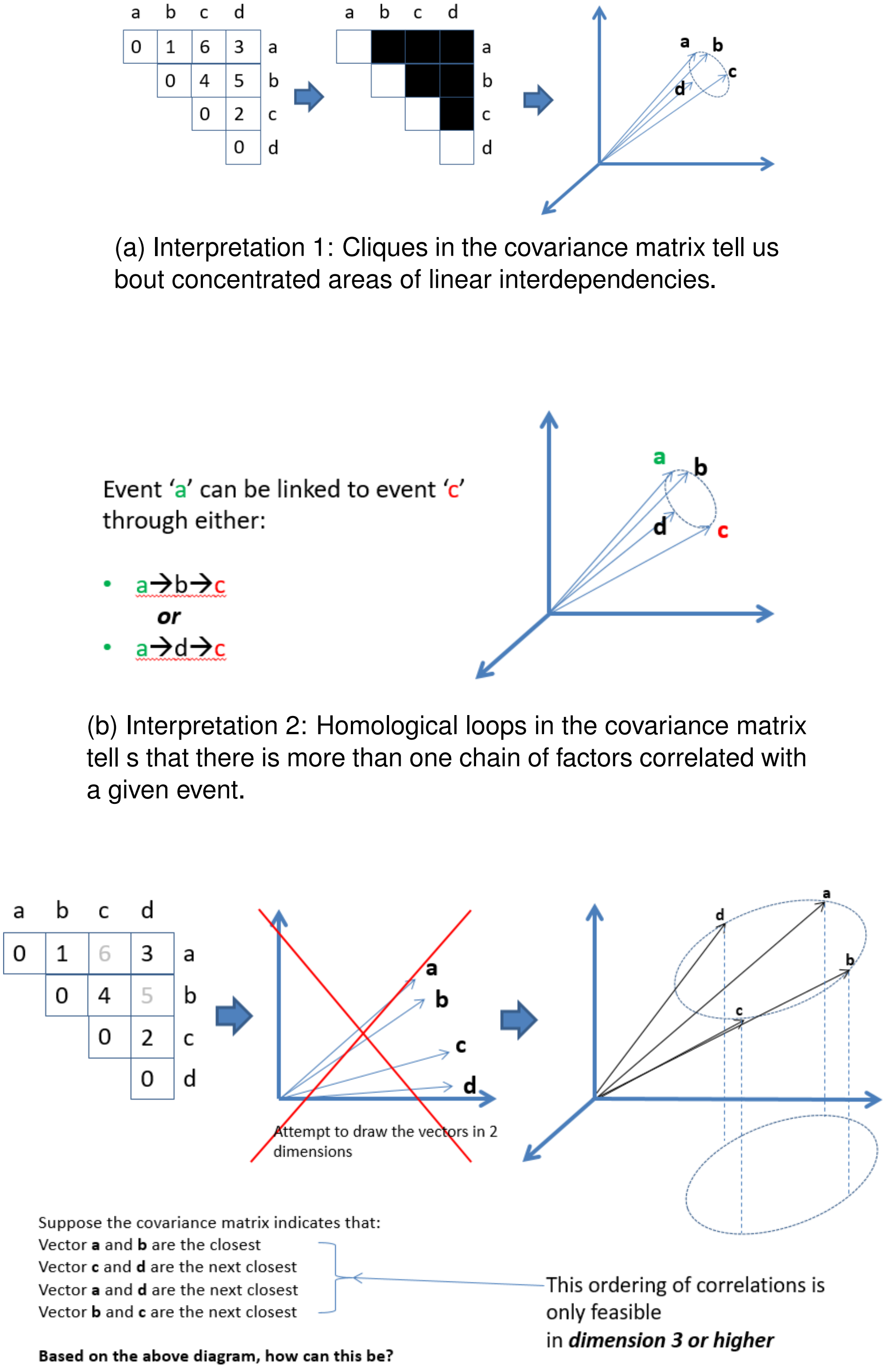
Update Graph

The 3-clique {(a,b), (b,d), (a,d)} is a 2-D cycle composed of Three 2-cliques

(d) Threshold Again

Figure 30: Basic Steps to Compute Clique Homology

Fig. 31 depicts three interpretations that can be made regarding a set of data vectors from a data matrix, as they relate to their corresponding covariance matrix. The first interpretation is that vectors that are all close to each other form a region of high correlation due to their interdependency. Next, loops or circuits that can be traveled from one data vector to another represent a chain of factors that can be (perhaps, but not necessarily) causally linked. Finally, the ordering of the covariance matrix entries (highly related to the angular separation between data vectors) is tightly coupled with the rank of the covariance matrix [26], a measure of the data sets intrinsic geometry in some sense.



a	b	c	d	
0	1	6	3	a
	0	4	5	b
		0	2	c
			0	d

(a) Interpretation 1: Cliques in the covariance matrix tell us bout concentrated areas of linear interdependencies.

Event ‘a’ can be linked to event ‘c’ through either:

- a→b→c

  or
- a→d→c

(b) Interpretation 2: Homological loops in the covariance matrix tell s that there is more than one chain of factors correlated with a given event.

a	b	c	d	
0	1	6	3	a
	0	4	5	b
		0	2	c
			0	d

Suppose the covariance matrix indicates that:
Vector **a** and **b** are the closest
Vector **c** and **d** are the next closest
Vector **a** and **d** are the next closest
Vector **b** and **c** are the next closest

This ordering of correlations is only feasible in ***dimension 3 or higher***

Based on the above diagram, how can this be?

(c) Interpretation 3: The ordering of covariance matrix entries (related to the rank of the matrix) constrains the intrinsic dimensionality of the geometry spanned by the data vectors.

Figure 31: Covariance Matrix Cliques, Loops, and Orderings

Appendix D: Sheehy Report

Homological Methods for Sensor Network Integrity

(Kirk Gardner and Donald R. Sheehy)

1 Introduction

In many settings, large networks of sensors collect data. We are studying methods to measure, analyze, visualize, summarize, and compare global behaviors of the network over time. Major challenges include sensor errors, gaps in coverage, and a changing network.

Homological Sensor Networks Our work has focused on the homological sensor network (HSN) setting. Also known as coordinate-free sensor networks, this model attempts to impose minimal local assumptions on the underlying network. It is well-suited to address the following challenging settings:

- The sensors do not have GPS or other coordinates. The do not know where they are, only which sensors are nearby (their neighborhood).
- The underlying domain the sensors are intended to cover is unknown. The sensors can detect if they are near the boundary of the domain, but that is all.
- Everything about the network can change from one time to the next including the positions and identities of the sensors. Only the underlying domain and the phenomenon they are measuring is assumed to be constant or changing continuously.

A naive approach to computing signatures or summaries of data collected from such a network, would be to compute statistics from the sensors, say the mean or variance of the collected data. This approach throws out the fundamental information about local neighborhoods that is inherent in the network and reflects the underlying domain. Surprisingly, it’s possible to have methods that both reflect shape information (from the local neighborhoods to the global network) and operate without coordinates or geometry.

HSNs address these challenges by integrating data collected locally using the neighborhood information into a single global signature called a persistence diagram. These signature are defined in terms of an algebraic topological invariant called (persistent) homology, from whence they derive the name. The homology characterizes the shape in a way that does not depend on how it is situated in space. The persistence diagram then captures aspects of the data that do not depend on positions of the sensors. The only requirement is that the sensors have sufficeint coverage.

Persistence diagrams are a natural choice for data analysis on sensor networks due to several desirable properties.

- The persistence diagram uses global shape features rather than coordinates or landmarks to summarize the data.
- The persistence diagram is naturally stable to perturbations in the data.
- The persistence diagram is invariant to continuous (invertible) changes in the underlying domain.

- The persistence diagram from a network with good coverage can be related directly to the ground truth on the domain, and so the specific choice of the sensor locations and the under-lying network have a negligible impact on the persistence diagram.
- Persistent homology can be used to verify coverage in the network. The Topological Coverage Criterion (TCC) of de Silva and Ghrist [1] uses persistent homology to verify coverage of an unknown domain by a coordinate-free sensor network, and was extended to weighted k-coverage in a more general setting in [2].

These advantages bring with them some new challenges that we addressed in this research. Because the signature is using both the measurements and the local neighborhoods defined by the network, there are very distinct reasons why two signatures might differ.The data could be different, the domain could be different, or the network could be different. As a result, the major thrust of our research looked at the relative magnitude of these differences. For example, can we tell if the persistence diagram changed because of a significant change in data or because of a significant change in the underlying network? These issues make it difficult to distinguish network anomalies from meaningful changes in the data.

Another major challenge we address is how to analyze networks that change over time. That is, we want to the characterize the behavior of a network in a way that allows us to compare the behavior of a network from one day to the next or to compare one networks measurements over time to that of another network. Constructing a summary of such an event that is both descriptive and discriminative is increasingly difficult when the network is allowed to change over time. Persistent homology is particularly well-suited to this setting as it provides a reliable summary of global behavior from local information alone. This summary is robust to missing data and stable under reasonable changes to the network.

Our initial experiments focused on evaluating the robustness of persistent homology to missing data and how it could be used to compare data sets. Persistent homology can be computed from a similarity matrix, such as a matrix of pairwise distances between points. In theory, unreliable data should be omitted, because it may appear as a significant feature in the persistence diagram. To this end, we conducted an experiment in which pairwise distances were removed from a sample of a 2-torus, a surface for which the persistent homology is known. We found that even when over 50% of the edges were removed the most significant features remained, with additional features resulting from the missing data presenting as additional noise.

Stability and its Consequences A fundamental feature of persistent homology is its stability to small changes in the network. As a result, we can compare diagrams of networks exhibiting similar behavior when they would otherwise be incomparable using traditional methods. We demonstrated one way to express the usefulness of this property using collections of parametric surfaces. The persistence diagrams of random samples of surfaces with different parameters, namely 2-tori with varying radii, were compared and embedded in low dimensional space. We found that the embed-ding reflected the parameter space from which the tori themselves were constructed, indicating that the persistent homology could be used to expose global structural differences. Moreover, because the metric on persistence diagrams focuses on the most prominent features our experiments on the robustness of persistence diagrams indicates that these these structural differences are still exposed in the presence of missing data.

Extending the HSN setting Current work has been focused on applying the observations made in these experiments to sensor network data. This differs from the traditional setting of coverage in homological sensor networks in a fundamental way. Previous work on coverage verification used a pair of neighborhood graphs constructed from sensor proximity data at two scales. Coverage is verified by the homology of the inclusion from one scale into another, which may be seen as a small subset of the persistent homology of the distance to the network. It has been shown that function values (i.e. measurements) on a collection of points which cover some unknown domain can be used to approximate the persistent homology of the function on the entire domain [3]. Once we have verified coverage, sensor measurements give a reliable signature of the measured event. Persistence stability allows us to aggregate signatures of events occurring over time even when the network is changing, provided coverage is maintained.

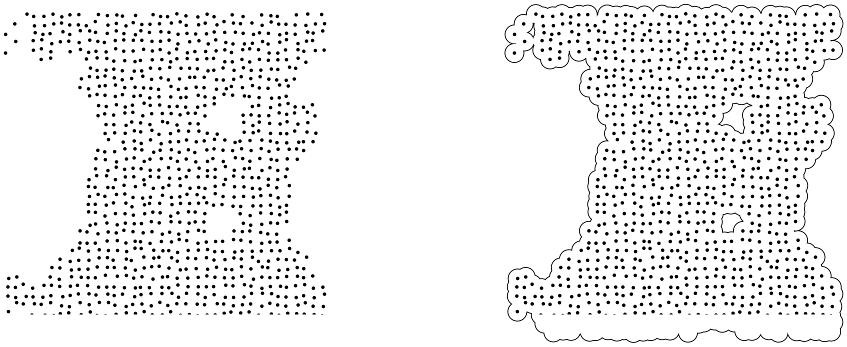


Figure 1: A collection of points representing sensors in some domain and the region they cover.

Lastly, verifying coverage requires that sensors can detect the boundary of the domain. We are currently investigating how this additional information can be used to compare networks covering distinct domains that observe a common event.

Extending the HSN setting Current work has been focused on applying the observations made in these experiments to sensor network data. This differs from the traditional setting of coverage in homological sensor networks in a fundamental way. Previous work on coverage verification used a pair of neighborhood graphs constructed from sensor proximity data at two scales. Coverage is verified by the homology of the inclusion from one scale into another, which may be seen as a small subset of the persistent homology of the distance to the network. It has been shown that function values (i.e. measurements) on a collection of points which cover some unknown domain can be used to approximate the persistent homology of the function on the entire domain [3]. Once we have verified coverage, sensor measurements give a reliable signature of the measured event. Persistence stability allows us to aggregate signatures of events occurring over time even when the network is changing, provided coverage is maintained.

Lastly, verifying coverage requires that sensors can detect the boundary of the domain. We are currently investigating how this additional information can be used to compare networks covering distinct domains that observe a common event.

2 Background

Topological Data Analysis (TOA) is an approach to studying potentially high-dimensional, incomplete, or noisy data by studying its "shape." The primary assumption is that this data carries relevant geometric and topological information about the "system" from which it has been generated. Homology is a powerful tool from algebraic topology which can be generally understood as a way to measure the components, loops, and voids in a space. Persistent homology is a popular tool in TOA which tracks the evolution of the homology of a space over a range of scales. For example, the homology of a sensor network can be used to detect holes in coverage at a given scale, where scale can be understood as the sensor's coverage radius.

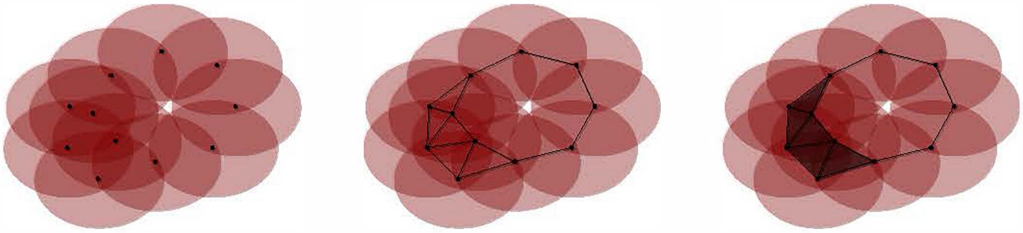


Figure 2: (Left) The coverage regions of a collection of points P at some scale α. (Middle) The neighborhood graph with edges for each pair of points within pairwise distance α. (Right) If we attempt to fill cycles in the graph with triangles we identify a cycle that cannot be filled which reflects 1-dimensional homological feature, a loop.

The key observation motivating the use of homology in coverage verification is that it can be computed without coordinate information. If a network's sensors can simply detect the presence within some radius a representative structure known as the (Vietoris-)Rips Complex can be com-puted from the resulting neighborhood graph, as shown in Figure 2. As this radius increases, holes in coverage may appear or disappear, giving a topological signature for the network as the sensor's coverage radius increases. The changes in the homology of the corresponding complex, as shown in Figure 3 (left), can be readily computed and reflect the changes in the coverage region itself. The resulting signature is known as a persistence diagram, depicted in Figure 3 (right).

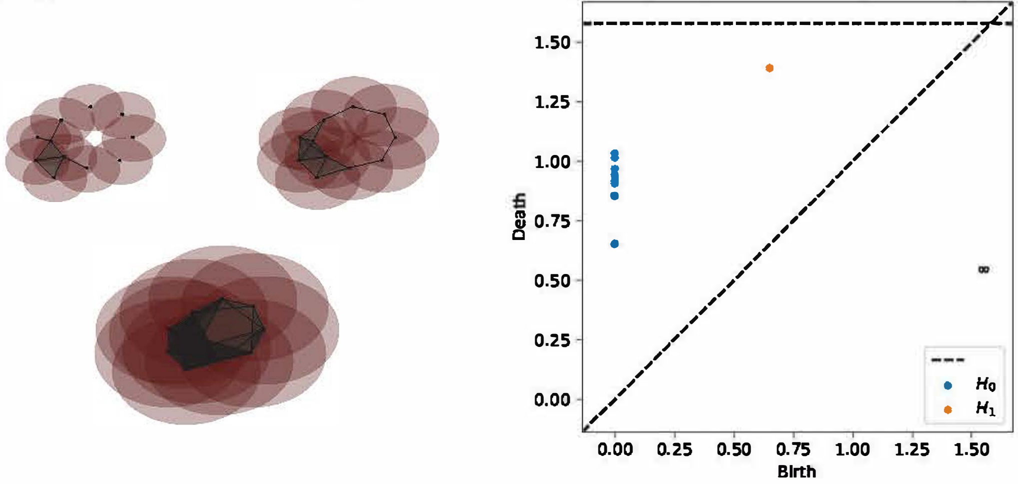


Figure 3: (Left) A filtration of rips complexes at scales 0.5, 0.7, and 1.4 illustrating a point (0.7, 1.4) in the corresponding 1-dimensional persistence diagram (right, orange). The 1-dimensional feature that is born at scale 0.7 persists until it dies at scale 1.4.

The space of persistence diagrams comes equipped with a metric known as the bottleneck dis-tance, depicted in Figure 4. VVe use the bottleneck distance extensively as a way to compare collections of closely related spaces, such as networks covering the same domain or phenomenon measured by different networks.

3 Research Results

3.1 Robustness to Incomplete Data

Interpreting sensor network data requires integrating information from multiple sources that may be prone to error. Although most sources may be functioning properly missing data has the potential to corrupt an entire dataset when analyzed with traditional methods. Topological methods, on the other hand, integrate local information in a way that is robust to missing data. This property was one of the original motivations for the use of persistent homology in the study of data. In particular, the stability theorem states that the persistent homology of the filtered Vietoris-Rips complex built from a point cloud is robust to slight perturbation of the points.

In order to test the extent of this stability we investigated the effect of gaps in the pairwise similarity matrix used to construct the complex. Such a situation may arise as the result of a missing, corrupted, or untrustworthy measurement, or as a result of intentional sparsification or subsampling of the similarity matrix.

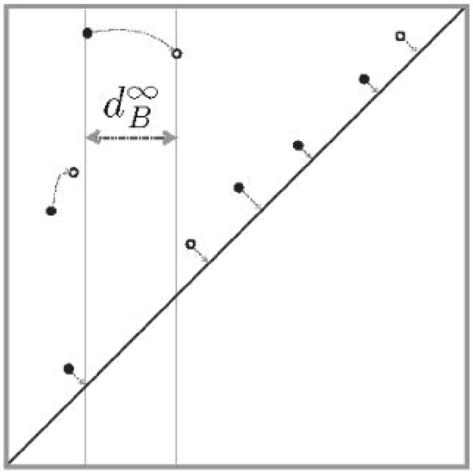


Figure 4: The bottleneck distance is given by a matching of points in two diagrams. Points that are considered noise may be matched with the diagonal.

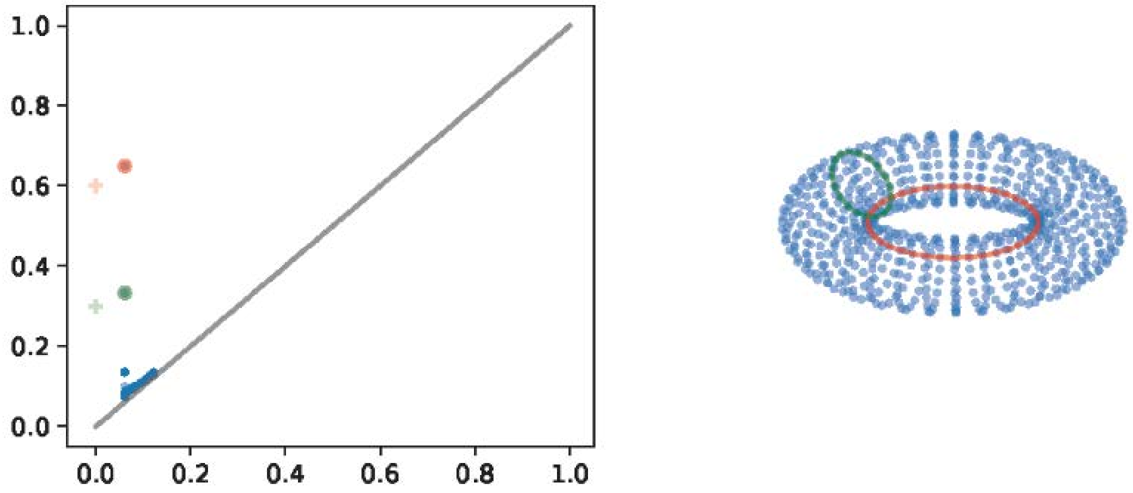


Figure 5: Persistent features of 722 uniformly spaced points sampled from $\mathbb{T}$. The green and red $+$ markers represent expected features at $r = 0.3$ and $R - r = 0.6$ respectively.

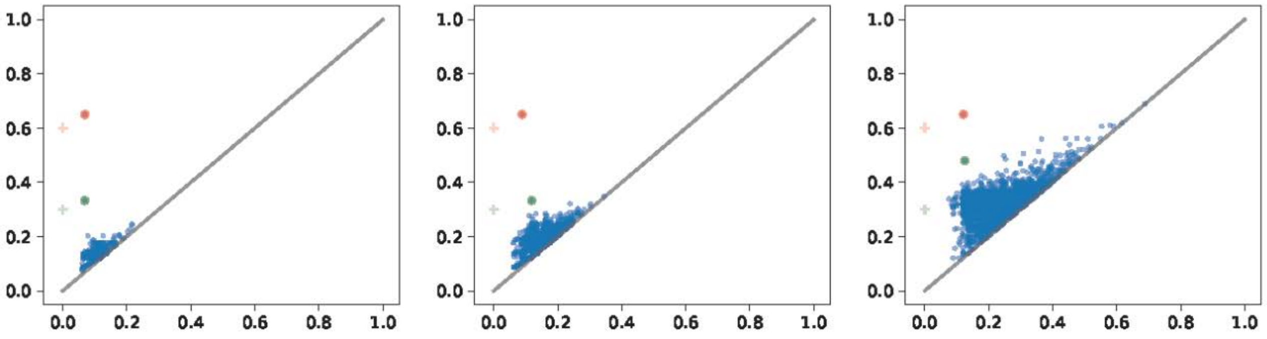


Figure 6: Persistence diagrams of 722 uniformly spaced points sampled from a 2-torus with 25%, 50%, and 75% of the edges removed at random.

3.2 Embeddings of Persistence Diagrams

Often data that is collected in some high dimensional space has a far lower intrinsic dimension. Moreover, the same event may be measured in different ways that renders two measurements incomparable, despite similar fundamental structure. Persistent homology has been shown to be a useful tool for studying point clouds that are sampled from a single unknown space in order to learn something about this structure. We instead considered collections of point clouds sampled from a family of related spaced defined by a so-called parameter space. Our hypothesis was that we could use the bottleneck distance to embed the diagrams of this family of spaces in order to learn something about the structure of the underlying parameter space. Specifically, we used Multidimensional Scaling (MOS) to embed collections of persistence diagrams in low-dimensional space.

We also formalized the notion of parametric families of spaces and attempted to characterize those with common topological structure. Namely, Tori, Spheres, and Klein bottles. Our goal was to relate this structure to that of the parameter space by first develooina a theorv on the soaces themselves.

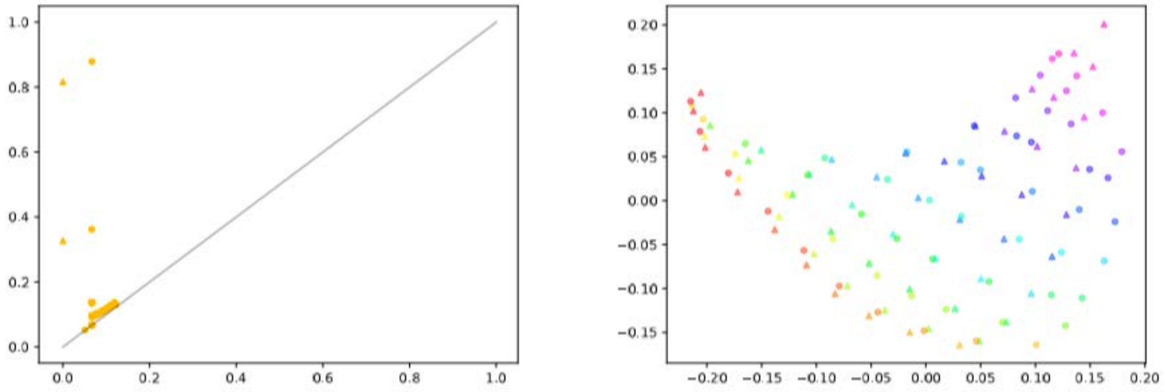


Figure 7: (Left) Ideal barcode (triangles) and sampled barcode (circles) of of a 2-torus parameterized by (0.2, 0.7). (Right) 2D MDS embedding of the pairwise bottleneck distances between 110 ideal (triangles) and sampled (circles) barcodes.

Figure 7 shows an ideal (triangle markers) and sampled (circle markers) persistence diagram, and the 20 MOS embeddings of a collection 110 diagrams with varying major and minor radii.

This suggests that diagrams of sampled tori may be used to recover their corresponding radii by comparison with their expected diagrams in the embedding.

This initial experiment on parametric families of shapes is closely related to our current research in a fundamental way. Instead of collections of closely related shapes, we use this same embedding technique to study collections of measurements by one or many networks. The resulting collection of persistence diagrams traces a collection of curves in persistence space in the same way a parametric family of tori traces out its parameter space. Our hope is that this technique can be used to quantify changes in a network, or a some phenomenon observed by the network in the same way it can be used to recover the parameters in this experiment.

3.3 Software Testbed

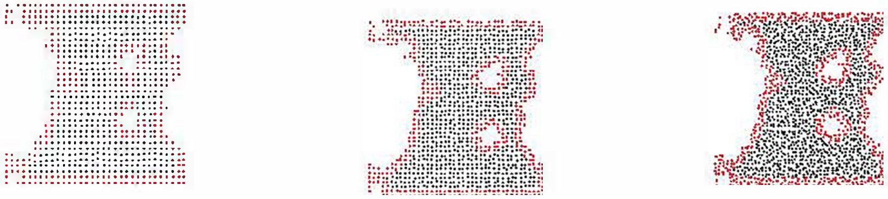


Figure 8: Multiple samples of the same domain with samples close to the boundary in red.

Our implementation of the TCC, which is detailed in our preliminary report, has been extended to the analysis of scalar fields over coordinate-free sensor networks. In particular, we have modified our software to generate domains can be sampled with varying resolution to simulate networks which cover the domain over a range of scales, as seen in Figure 8. Assume we are given such a sequence of networks in addition to a corresponding sequence of sensor measurements taken as a subset of the values of some function over time defined on the domain. We demonstrate a novel signature these time-varying functions that extends the tools used in the TCC to the analysis of scalar fields over time.

Our previous experiments focused on the persistent homology of Rips filtrations constructed from point clouds in euclidean space. This filtration is closely related to the sequence of metric balls growing around the point cloud as the persistent homology of both gives a signature for distance to the point cloud. In particular, the persistent homology of the Rips complex of a point cloud approximates that of the distance to the point cloud as a function on the underlying metric space.

More generally, the persistent homology of a real-valued function captures the changes in the homology groups of its sublevel-sets-the set of paints with function values below a given scale. The distance to a point cloud is a real-valued function with sublevel-sets equal to the union of metric balls at a given scale. Under certain conditions we can approximate the persistent homology of a real-valued function given only its values on a finite subset of its domain. Figure 9 depicts a function on a subset of the plane and its function values on the simplices of a simplicial complex defined on a subset of the plane. The filtration given by ordering the simplices of this complex by their function values is known as an induced filtration. Ghazal et. al. detail how to construct a filtration.

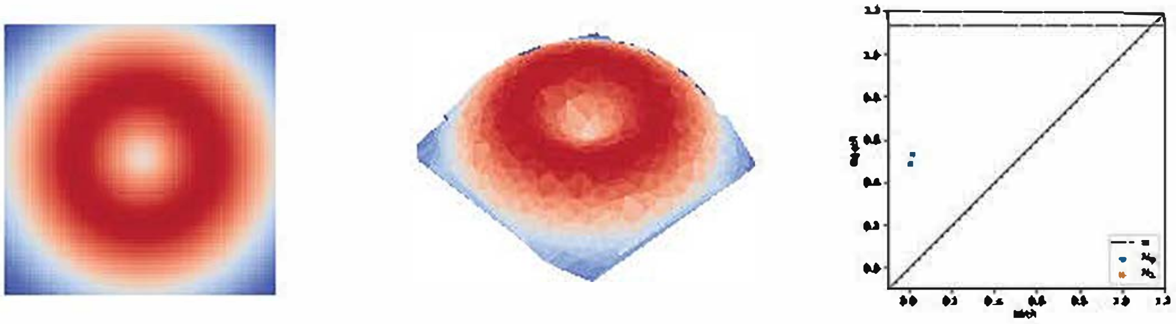


Figure 9: (Left) Continuous function on the plane. (Middle) Function values (as z-axis) on a random sample extended to a simplicial complex. (Right) Persistence diagram of an induced filtration.

The inclusion of two filtrations) that approximates the persistence diagram of the function itself [3] that has a natural application to measurements by coordinate-free sensor networks. Extending our testbed to explore this result was a natural next step as it assumes coverage and the structures required are a subset of those used in the computation of the TCC. This extension has led to promising results on applications to functions on coordinate-free networks over time as well as interesting theoretical questions on the role of the boundary in these experiments .

A coordinate-free sensor network defined on an unknown, bounded domain consists of a nested pair of neighborhood graphs as well as a nested pair of subgraphs. The vertices of the neighborhood graphs correspond to sensors in the network with edges representing the presence of nearby sensors at two scales. The subgraphs are the restriction of the two neighborhood graphs to sensors close to the boundary of the domain. Moreover, we assume that sensors can measure the value of a real-valued Lipschitz function defined the domain. A measurement made by a sensor corresponds to the function value at the location of that sensor, which remains unknown. Similarly, a sensor network is said to cover a domain at a given scale if the union of metric balls centered at the locations of the sensors includes the domain at that scale.

The proposed setting is a dynamic collection of sensors. Sensors can be moved, added, or deleted but always maintain coverage of the under1ying domain at some scale. An event that occurs on the domain is captured by the sensors over time in a sequence of measurements by distinct networks.

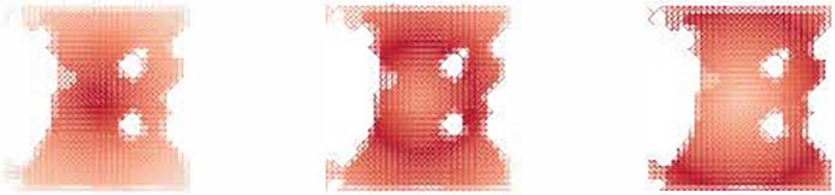


Figure 10: Function values on a single network over time (top row) and their corresponding persistence diagrams.

That is, we are given a sequence of coordinate-free sensor networks which all cover some un-known. bounded domain. Each network in the sequence corresponds to a step in time and measures.

a real-valued Lipschitz function defined on the domain. Rips complexes are constructed from the provided neighborhood graphs and their simplices are reordered to form a sequence of nested pairs of Rips complexes. As in [3] the persistent homology of this sequence of nested pairs approximates that of the function within a factor proportional to the coverage radius. The resulting sequence of persistence diagrams therefore approximates that of the function over time and traces a curve in persistence space, referred to as a trajectory signature, as seen in Figure 12.

4 Trajectory Signatures

Throughout, the time-varying function used is the distance to a ring in the plane with increasing radius. The function values on simplices of a network are depicted by color in Figure 10. Reordering these simplices by function value gives a filtration for each time step.

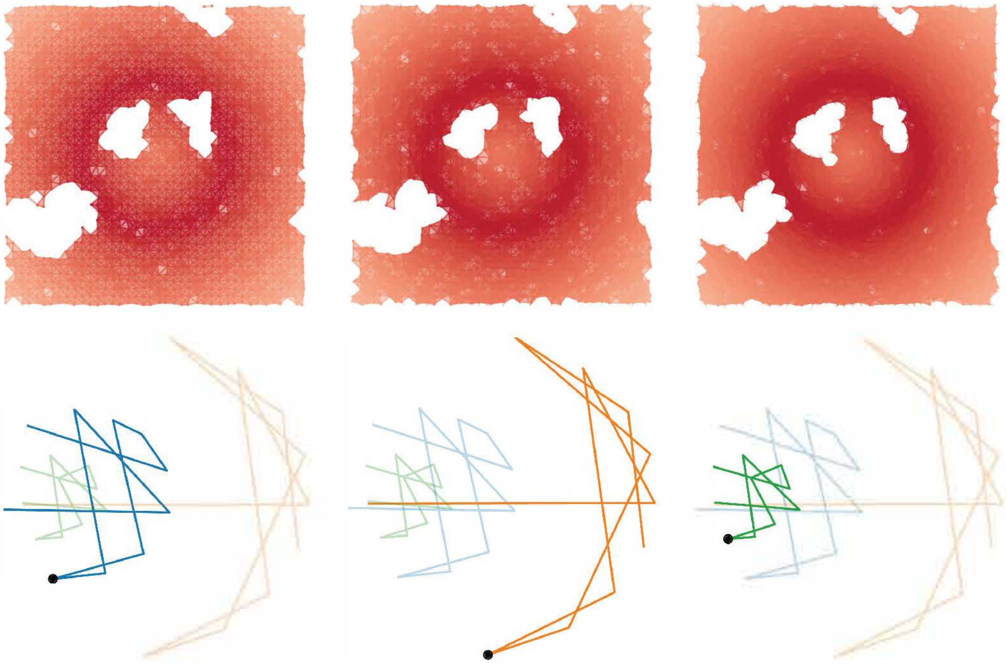


Figure 11: (Top) Function values on three different networks that cover the same domain with at a single time step. (Bottom) Trajectories of the function on the networks computed without boundary information. The black the point corresponds to the point on the curve corresponding to the time step shown.

When is information about the boundary needed? Our software was first applied to time-varying functions on domains without any information about the boundary of the domain. Although the resulting signatures were sensitive to changes in the function we found that they were too sensitive to features of the network. Moreover, the trajectories did not appear to reflect the simple structure of the function as demonstrated in previous experiments.This can be seen in Figure 11 which depicts three curves of the same function on three different networks covering the same domain. The curves represent three very different signatures as a result of noise in the individual networks and a general lack of persistent features due to obstruction by features of the domain. Because information about the boundary is required in order to verify coverage using the

TCC we then considered how this additional information could be used to account for known features of the domain.

What do we learn when the boundary is needed? Our algorithm was modified to use information about the boundary of the domain instead computing the persistent homology of the function relative to the restriction to the boundary. Our hope was that, by accounting for known features of the domain captured by the boundary we could direct the focus of the signature to the features of the function. The results, as depicted in Figure 12, show that this modification yields signatures that are stable to changes in the domain while also reflecting the behavior of the observed function. The curves seem to follow a common path, unlike those in Figure 11, indicating that information about a common boundary yields a signature stable to changes in the network. However, it is possible that these signatures are a result of too much dependance on the function and tell us nothing about the domain.

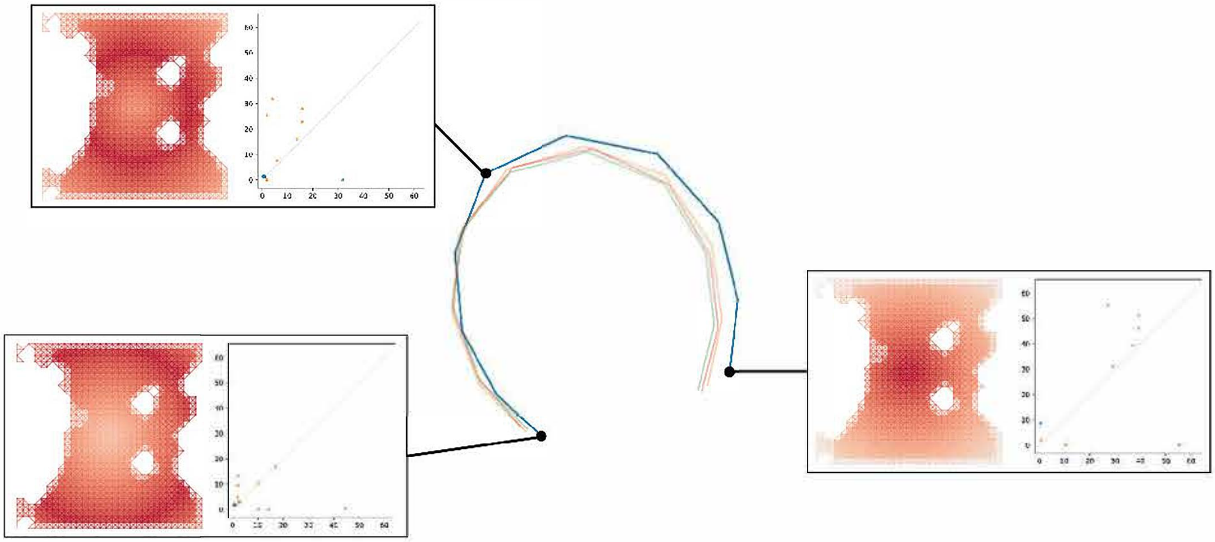


Figure 12: A collection of overlaid trajectories from different networks. For three time steps of one of these trajectories the function values on the corresponding networks and resulting persistence diagrams are shown.

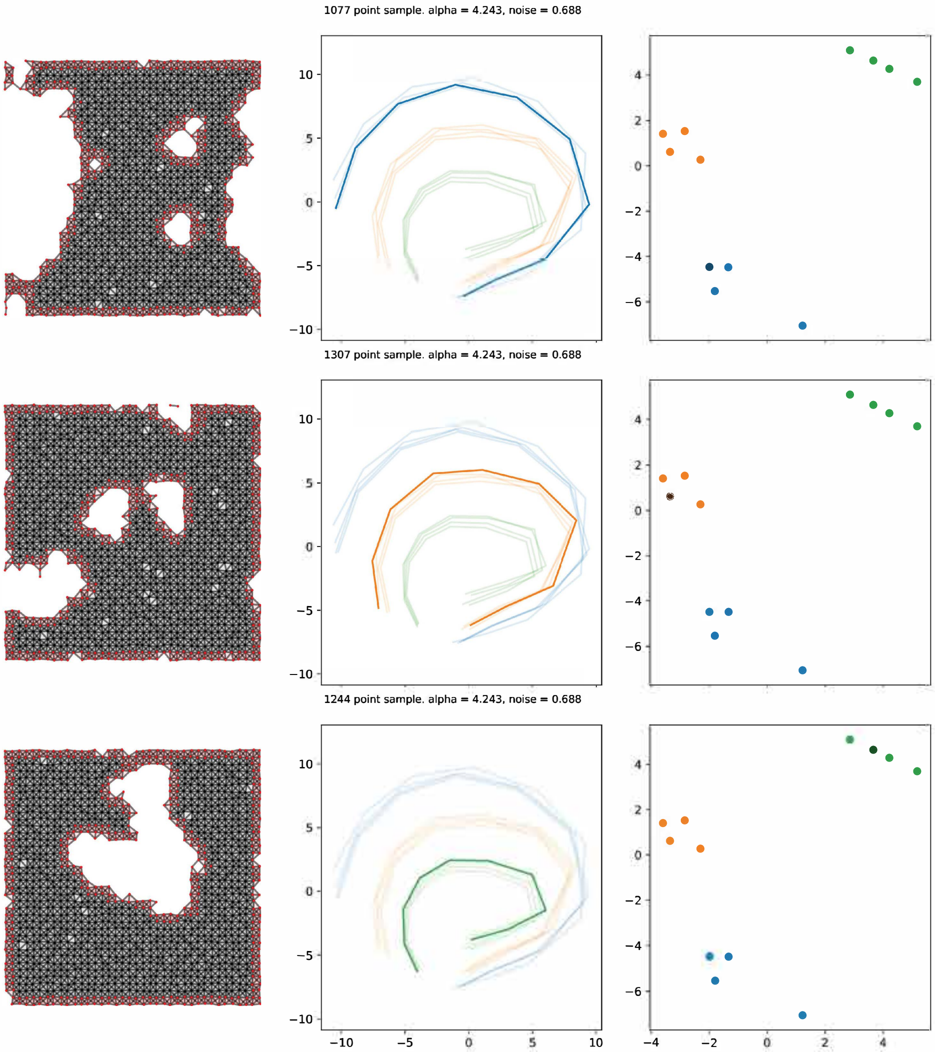


Figure 13: (Left column) networks constructed from samples of 3 distinct domains corresponding to blue (top), orange (middle), and green curves/points in the following. (Middle column) A low-dimensional embedding of curves drawn by the same function on each domain. (Right column) An embedding of the maximum distance between curves for each network.

What can we learn about the domain? To eliminate the possibility that this signature is dominated by the topology of the function we applied the same process to the same function applied to multiple domains. We found that the resulting trajectories clustered into distinct groups, indicating that changes in the domain are clearly reflected, as seen in Figure 13. This suggests that the signature is sufficiently discriminative, and may therefore be useful for detecting changes in the domain.

5 Lessons Learned and Future Directions

Through our preliminary experiments we discovered a natural application of persistent homology in the intersection of homological sensor networks and the analysis of time-varying functions. In particular, we considered an event observed by a network that covers some unknown domain that is changing over time. Each measurement of the event by the network, formalized as a sampleof some function on the domain, is integrated and summarized by the persistence diagram of the function as "seen" by the network. This allows us to compare global behaviors of the network over time in a way that is robust to missing data as well as changes in the network.

We found that applying the machinery used to confirm coverage by a coordinate-free network can be applied to the analysis of scalar fields on these networks. In particular, we propose a signature for time-varying functions on coordinate-free networks that is stable to changes in the network.

The extension of scalar field analysis to bounded domains alone is an interesting direction for future theoretical work. Preliminary work indicates that this requires formalizing fundamental results in algebraic topology to the persistent homology of functions on pairs of spaces. This work may lead to useful theoretical guarantees assuming the conditions in [2]. Another future direction is to look at more interesting time-varying functions such as heat flow and real-world sensor data. Lastly, Figure 13 indicates potential for these signatures as a way to identify changes in the domain or as tool for the "fuzzy" classification of spaces parameterized by a function.

References

[1] Vin de Silva and Robert Ghrist. Coverage in sensor networks via persistent homology. Algebraic & Geometric Topology, 7:339–358, 2007.

[2] Nicholas J. Cavanna, Kirk P. Gardner, and Don Sheehy. When and why the topological coverage criterion works. In SODA, 2017.

[3] F. Chazal, L. J. Guibas, S. Y. Oudot, and P. Skraba. Analysis of scalar fields over point cloud data. In Proc. 19th ACM-SIAM Sympos. on Discrete Algorithms, pages 1021–1030, 2009.

LIST OF SYMBOLS

Symbol	Meaning
$\|C\|_F$	Frobenius Norm of the Matrix C
$\|C\|_2$	Two-Norm of the Matrix C
$\|C\|_*$	Nuclear Norm of the Matrix C
β_i	i^{th} Betti Number
$\mathcal{H}$	Shannon Entropy of Singular Values Structure Statistic
ϕ_{ij}	Entry i, j of the Sensing Matrix Φ
σ_i	i^{th} Singular Value
Δ	Generic Designation for Difference between Expected and Estimated Features
Σ	Diagonal Matrix of Singular Values in Singular Value Decomposition
Φ	Sensing Matrix Used in Random Hadamard Statistic
a	Parameter Used in a Non-Informative Matrix N_a
c_{ij}	i^{th}, j^{th} entry of the Covariance Matrix C
d_{EUC}	Euclidean (Frobenius) Distance between two Covariance or Correlation Matrices
d_N	Distance to Nearest Non-Informative Matrix N
f	A Set of Normalized Singular Values
h	Random Hadamard Structure Statistic
i, j	Generic Indices Used in Vectors and Matrices
C_i	i^{th} Correlation or Covariance or Gram Matrix
D_i	i^{th} Data Matrix
$E[]$	Expectation Operator
$F(x)$	EmpGeneric designation for an Estimated Feature Set Extracted from Data
F_{est}	Generic designation for an Estimated Feature Set Extracted from Data
F_{exp}	Generic designation for an Expected Feature Set Extracted from Data
H	Centering Matrix
$H(f)$	Shannon Entropy of the set of Normalized Singular Values f
I	Identity Matrix
$\mathtt{I}(\cdot)$	Indicator Function
J	Constant Ones Matrix
M	Number of rows or Records of a Data Matrix
N_a	A Non-Informative Matrix of Parameter a
N_{triu}	The Upper Triangular Entries of a Symmetric Matrix
N	Number of columns or Dimensions of a Data Matrix D
$tr(A)$	Trace Operator of the Matrix A
U	Set of Left-Singular Vectors in Singular Value Decomposition
V	Set of Right-Singular Vectors in Singular Value Decomposition
X_i	A Single i^{th} Discrete Probability Mass Function
X	A Collection of Discrete Probability Mass Functions X_i

LIST OF ACRONYMS

Abbreviation	Meaning
DR	Dimensionality Reduction
DSA	Dynamic Spectrum Access
EVD	Eigenvalue Decomposition
GDB	Geographic Database
ICA	Independent Component Analysis
KLD	Kullback-Leibler Divergence
KS	Kolmogorov-Smirnov Statistic
LS	Least Squares
MDS	Multi-Dimensional Scaling
NNMF	Non-Negative Matrix Factorization
PCA	Principal Component Analysis
PSD	Power Spectral Density or Positive Semi-Definite (Matrix)
RF	Radio Frequency
SA	Sparse Approximation
SAS	Spectrum Access System
SN	Sensor Node
SNR	Signal to Noise Ratio
SPD	Symmetric Positive Definite Matrix
SSDF	Spectrum Sensing Data Falsification
SVD	Singular Value Decomposition
TCC	Topological Coverage Criterion
TDA	Topological Data Analysis
TOPEX	Topological Exploitation